

Санкт-Петербургский Государственный Университет

Поляков Иван Михайлович

Выпускная квалификационная работа

Адаптивный метод глобальной оптимизации липшицевых функций

Уровень образования: магистратура

Направление 01.04.02: «Прикладная математика и информатика»

Образовательная программа ВМ.5505.2021

«Математическое и информационное обеспечение экономической деятельности»

Научный руководитель:

доктор физ.-мат. наук,

профессор Крылатов Александр Юрьевич

Рецензент:

инженер ключевых проектов,

ООО «Техкомпания Хуавей»,

кандидат физ.-мат наук,

Коврижных Николай Александрович

Санкт-Петербург

2023 г.

Содержание

Введение	3
Обзор литературы	4
Постановка задачи	7
1 Глобальная оптимизация липшицевых функций	8
2 Оптимизация с фиксированной константой Липшица	15
2.1 Алгоритм	15
2.2 Анализ сходимости	17
3 Оптимизация с неизвестной константой Липшица	19
3.1 Алгоритм	19
3.2 Анализ сходимости	20
4 Модификация метода оптимизации с неизвестной константой Липшица	23
4.1 Алгоритм	23
5 Результаты	24
Заключение и выводы	28

Введение

Целью данной работы является разработка последовательных стратегий, которые приводят к эффективной оптимизации неизвестной функции при единственном предположении, что она имеет конечную константу Липшица. Сначала определяются достаточные условия непротиворечивости общих последовательных алгоритмов и формулируется их ожидаемая минимальная эффективность. Затем анализируется алгоритм *LIPO*, который предполагает, что константа Липшица известна. Также представлена адаптивная версия этого алгоритма, когда константа Липшица неизвестна и её необходимо оценивать в процессе оптимизации. Наконец, предложен и проанализирован новый алгоритм, где также неизвестна константа Липшица, но изменена стратегия принятия решения для выбора следующей точки вычисления.

Обзор литературы

С быстрым развитием компьютерных технологий высокоточное моделирование стало незаменимым инструментом в современных отраслевых приложениях, которое может эффективно сократить бюджет проектирования и принести более высокие экономические выгоды [1], [2], [3], [4]. В то же время, когда требования к точности постоянно возрастают, вычислительные затраты на анализ моделирования могут стать огромными, что вызовет трудности в разработке методов оптимизации [5], [6], [7], [8]. Кроме того, многие реальные приложения, такие как управление, планирование, логистика, проектирование структур и распознавание образов, включают в себя дискретную оптимизацию [9], [10], [11] и времязатратный анализ моделирования [12]. Таким образом, дискретные и ресурсоемкие задачи глобальной оптимизации являются сложными и в последние годы начинают привлекать все больше внимания [13].

Для задач дискретной оптимизации алгоритм *Branch and Bound (BB)*, который рекурсивно делит множество решений и оценивает граничные значения, может найти оптимальную комбинацию этих дискретных значений [14]. Например, процедуру этого алгоритма использовали для планирования нескольких проектов с ограниченными ресурсами [15], а его модификация применялась для выбора оптимальных подмножеств признаков в приложениях распознавания образов [16]. Однако данный алгоритм является неподходящим для задач глобальной оптимизации, требующих больших вычислительных ресурсов, потому что он строит упрощенную задачу, глобальный оптимум которой должен быть найден для определения нижней границы, из-за чего возникнет множество вызовов дорогостоящих функций, особенно для мультимодальных задач.

Алгоритм *Variable Neighborhood Search (VNS)* является эффективным инструментом для глобальных задач комбинаторной оптимизации [17]. *VNS* может систематически исследовать возможные соседние структуры для определения локальных оптимумов, а затем находить глобальный оптимум с помощью возмущения. Алгоритм широко применяется в различных областях, таких как искусственный интеллект, кластерный анализ, планирование и т. д. [18], [19], [20]. *VNS* был в первую очередь разработан для задач целочисленной оптимизации с ограничениями на блоки, но его нельзя напрямую использовать для нелинейных задач с ограничениями.

Nonsmooth Optimization by Mesh Adaptive Direct Search (NOMAD) была разработана для задач оптимизации, требующих больших вычислительных ресурсов, и задач оптимизации «черного ящика» [21]. *NOMAD* — это метод оптимизации без производных, который применим для областей непрерывного, целочисленного и смешанного проектирования. Кроме того, *NOMAD* хорошо справляется с задачами нелинейной оптимизации с ограничениями, что делает его пригодным для большинства реальных приложений. Тем не менее, нет обширных численных исследований возможностей *NOMAD* в решении задач оптимизации, требующих значительных вычислительных ресурсов. Стоит отметить, что в актуальной на сегодняшний день литературе существует другой тип алгоритмов для решения дискретных задач и задач глобальной оптимизации методом «черного ящика», а именно эволюционные стратегии [22], [23], [24]. Как правило, алгоритмы, основанные на эволюции, вдохновлены каким-либо природным явлением и могут генерировать популяцию в каждом цикле для случайного поиска в проектном пространстве. С обновленной популяцией и многократной оценкой целевой функции можно постепенно получать многообещающие решения.

Большинство этих дискретных метаэвристических алгоритмов применялись в реальных приложениях. Например, был предложен алгоритм роя дискретных частиц – *Discrete Particle Swarm Optimization (DPSO-PDM)* для обнаружения сообщества в сложной сети [25]. *DPSO-PDM* переопределяет скорость и положение частиц и добавляет эволюционную операцию дискретизации, чтобы избежать попадания в локальные оптимумы.

Для облегченного проектирования больших стальных рам под фактической нагрузкой был разработан дискретный алгоритм, основанный на эволюции летучих мышей [26]. В процессе оптимизации элементы рамы были выбраны из дискретного набора стальных профилей, чтобы сформировать оптимальную конструкцию с учетом ограничений по прочности и перемещению.

Не так давно в работе [27] представили и сравнили два вдохновленных космологией алгоритма: *Improved Black Hole (IBH)* и *Multiverse (IMV)*, и применили их для дискретной оптимизации размеров структур. Были использованы три типа корпусов, включая стальные каркасы, стальные рамы и железобетонные рамы, чтобы продемонстрировать их высокую эффективность. Хотя эти вышеприведенные метаэвристические алгоритмы могут эффективно решать проблемы «черного ящика» и дискретной глобальной оптимизации, они чрезмерно зависят от оценок функций, что может вызвать непосильную вычислительную нагрузку для трудоемких моделей [28], [29].

Оптимизация с помощью суррогатных моделей (*SAO*) сыграла важную роль в инженерных приложениях, основанных на имитационном моделировании, поскольку она довольно эффективна для вычислительно ресурсоемких задач [30], [31], [32]. Методы суррогатного моделирования, такие как *Kriging* [33], *Radial Basis Functions* [34], *Polynomial Response Surface* [35] могут эффективно организовать полученные данные для прогнозирования потенциальных решений, значительно уменьшая количество дорогостоящих оценок функций [36]. Однако большая часть существующей литературы по данному вопросу делает акцент на методах непрерывной области проектирования и редко сосредотачивается на дискретных случаях.

Существует большое количество алгоритмов, основанных на различных эвристиках, которые были введены для решения проблемы глобальной оптимизации, таких как генетические алгоритмы [37], методы на основе моделей (например, *Density Estimation* или *Discriminant Analysis* [38]) или Байесовская оптимизация [39]. В данной работе мы сосредоточимся на подходе к глобальной оптимизации, предполагая гладкость неизвестной функции. Этот подход основан на простом наблюдении, что во многих приложениях система демонстрирует некоторую регулярность по отношению ко входным данным. В частности, использование константы Липшица, впервые предложенное в основополагающих работах [40], [41], инициировало активное направление исследований и сыграло важную роль в разработке многих эффективных алгоритмов глобальной оптимизации, таких как *DIRECT* [42], *Multilevel Coordinate Search* [43] или *Simultaneous Optimistic Optimization* [44]. Свойства сходимости методов глобальной оптимизации были развиты в работах [45], [46] при предположениях о локальной гладкости, но, насколько известно, такие свойства не рассматривались в случае, когда указана только глобальная гладкость функции. Интересен также вопрос, насколько глобальные предположения о регулярности, которые в некотором смысле покрывают локальные, могут улучшить сходимость последних.

В данной работе решаются следующие вопросы:

1. найти ограничения и наилучшую производительность, которые могут быть достигнуты любым

алгоритмом над классом липшецевых функций;

2. разработать эффективный оптимальный алгоритм для этого класса задач.

Вклад текущей работы по отношению вышеупомянутых составляет следующие выкладки. Во-первых, здесь представлен новый алгоритм глобальной оптимизации, который использует глобальную гладкость функции и показывает хорошую производительность в тестах для оптимизации. Во-вторых, показано, что данный алгоритм может достигать более высоких скоростей сходимости на глобально гладких задачах, чем ранее известные методы, которые используют только локальную гладкость функции.

Постановка задачи

Пусть $\mathbf{X} \subset \mathbb{R}^d$ – не пустое, компактное и выпуклое множество, и пусть $f : \mathbf{X} \rightarrow \mathbb{R}$ – неизвестная функция, которая допускает максимум только во входной области \mathbf{X} . Цель глобальной оптимизации – найти такую точку x^* , что

$$x^* \in \arg \max_{x \in \mathbf{X}} f(x)$$

с минимальным количеством вычислений функции. Стандартный алгоритм включает в себя последовательную процедуру, которая начинается с вычисления функции $f(X_1)$ в начальной точке X_1 , и затем на каждом шаге $t \geq 1$ выбирается точка $X_{t+1} \in \mathbf{X}$, на основании предыдущих вычислений $(X_1, f(X_1)), \dots, (X_t, f(X_t))$, в которой вычисляется значение неизвестной функции $f(X_{t+1})$. После n итераций алгоритм возвращает точку X_{i_n} , где $i_n \in \arg \min_{i=1 \dots n} f(X_i)$, которая имеет наибольшее значение f . Эффективность алгоритма оценивается по разнице между глобальным максимумом функции и наибольшим полученным значением f :

$$\max_{x \in \mathbf{X}} f(x) - \max_{i=1 \dots n} f(X_i).$$

В анализе, представленном в данной работе, учитывается, что количество точек n не является фиксированным, и что функция f детерминирована. Более того, предположение о неизвестной функции f на протяжении всей работы состоит в том, что она имеет конечную константу Липшица:

$$\exists k \geq 0 : |f(x) - f(x')| \leq k \cdot \|x - x'\|_2, \quad \forall (x, x') \in \mathbf{X}^2.$$

Прежде чем приступить к анализу, стоит отметить, что аналогичная постановка задачи изучалась в работах [46] и [47], а также в [45] и [48] рассматривали зашумленные значения f .

1 Глобальная оптимизация липшицевых функций

Для каждого $x = (x_1, \dots, x_d) \in \mathbb{R}^d$ введем следующие обозначения.

Стандартная l_2 -норма:

$$\|x\|_2 = \left(\sum_{i=1}^d x_i^2 \right)^{-\frac{1}{2}}.$$

Шар с центром в точке x и радиуса $r > 0$:

$$B(x, r) = \{x' \in \mathbb{R}^d : \|x - x'\|_2 \leq r\}.$$

Для любого ограниченного множества $\mathbf{X} \subset \mathbb{R}^d$ определим его *радиус*

$$rad(\mathbf{X}) = \max \{r > 0 : \exists x \in \mathbf{X} | B(x, r) \subseteq \mathbf{X}\},$$

его *диаметр*

$$diam(\mathbf{X}) = \max_{(x, x') \in \mathbf{X}^2} \|x - x'\|_2,$$

а также его *объем* как $\mu(\mathbf{X})$, где $\mu(\cdot)$ есть мера Лебега.

Вдобавок,

$$Lip(k) = \{f : \mathbf{X} \rightarrow \mathbb{R} : \|f(x) - f(x')\| \leq k \cdot \|x - x'\|_2, \forall (x, x') \in \mathbf{X}^2\}$$

обозначает класс k -липшицевых функций, определенных на \mathbf{X} , а

$$\bigcup_{k \geq 0} Lip(k)$$

обозначает множество липшицевых непрерывных функций.

$\mathbb{U}(\mathbf{X})$ – равномерное распределение в ограниченной области \mathbf{X} ; $\mathbb{B}(p)$ – распределение Бернулли параметра p ; $\mathbb{I}\{\cdot\}$ – индикаторная функция, принимающая значения из $\{0, 1\}$ и обозначение $X \sim \mathbb{P}$ означает, что случайное значение X имеет распределение \mathbb{P} .

Чтобы разработать эффективные методы, сначала исследуем наилучшую производительность, которая может быть достигнута любым алгоритмом в классе липшицевых функций. Поэтому сначала опишем рассматриваемые здесь последовательные процедуры и соответствующее понятие согласованности в смысле глобальной оптимизации.

Определение 1.1. Класс алгоритмов оптимизации \mathcal{A} , который содержит все алгоритмы $A = \{A_t\}_{t \geq 1}$, может быть описан следующим образом:

1. Распределение A_1 , принимающее значение из \mathbf{X} и позволяющее сгенерировать первую точку вычисления, то есть $X_1 \sim A_1$;
2. Бесконечный набор параметрических распределений $\{A_t\}_{t \geq 2}$, принимающих значения из \mathbf{X} и основанных на предыдущих вычислениях, определяют итерационный цикл, то есть

$$X_{t+1} | X_1, \dots, X_t \sim A_{t+1}((X_1, f(X_1)), \dots, (X_t, f(X_t)))$$

Отметим, что данный класс алгоритмов также содержит детерминированные методы, в которых распределение $\{A_t\}_{t \geq 1}$ вырождено.

Определение 1.2. Алгоритм A глобальной оптимизации называется согласованным на множестве \mathbb{F} вещественных функций, допускающих максимум во входной области тогда и только тогда, когда

$$\forall f \in \mathbb{F}, \quad \max_{i=1\dots n} f(X_i) \xrightarrow{P} \max_{x \in \mathbf{X}} f(x),$$

где X_1, \dots, X_n обозначают последовательность n точек последовательности, сгенерированной алгоритмом A для функции f .

Теперь исследуем минимальные условия, при которых согласованный метод достигает асимптотической сходимости. Ожидается, что алгоритм глобальной оптимизации должен быть согласованным, по крайней мере, для класса липшицевых функций, и следующий результат выявляет необходимое и достаточное условие для этого случая.

Утверждение 1.1. Алгоритм A глобальной оптимизации согласуется на множестве липшицевых функций тогда и только тогда, когда

$$\forall f \in \bigcup_{k \geq 0} Lip(k), \quad \sup_{x \in \mathbf{X}} \min_{i=1\dots n} \|X_i - x\|_2 \xrightarrow{P} 0.$$

Доказательство. (Достаточность) Пусть A – алгоритм глобальной оптимизации, для которого выполняется сходимость из условия. Возьмем $\epsilon > 0$, любую функцию $f \in \bigcup_{k \geq 0} Lip(k)$ и пусть $\mathbf{X}_\epsilon = \{x \in \mathbf{X} : f(x) \geq \max_{x \in \mathbf{X}} f(x) - \epsilon\}$ будет соответствующим набором линий уровня. Так как \mathbf{X}_ϵ не пусто, тогда обязательно существуют такой $x_\epsilon \in \mathbf{X}$ и такой $r_\epsilon > 0$, что $B(x_\epsilon, r_\epsilon) \cap \mathbf{X} \subseteq \mathbf{X}_\epsilon$. Таким образом, если X_1, \dots, X_n обозначают последовательность из n точек, сгенерированная алгоритмом A для функции f , непосредственно получим из сходимости по вероятности следующее:

$$\begin{aligned} \mathbb{P} \left(\max_{x \in \mathbf{X}} f(x) - \max_{i=1\dots n} f(X_i) > \epsilon \right) &= \mathbb{P} \left(\bigcap_{i=1}^n \{X_i \notin \mathbf{X}_\epsilon\} \right) \\ &\leq \mathbb{P} \left(\bigcap_{i=1}^n \{X_i \notin B(x_\epsilon, r_\epsilon)\} \right) \\ &= \mathbb{P} \left(\min_{i=1\dots n} \|X_i - x_\epsilon\|_2 \geq r_\epsilon \right) \\ &\leq \mathbb{P} \left(\sup_{x \in \mathbf{X}} \min_{i=1\dots n} \|X_i - x\|_2 \geq r_\epsilon \right) \xrightarrow{n \rightarrow \infty} 0. \end{aligned}$$

(Необходимость) Пусть A – алгоритм глобальной оптимизации, согласующийся со множеством липшицевых функций, и предположим, что существует некоторая функция $f^* \in \bigcup_{k \geq 0} Lip(k)$ такая, что

$$\sup_{x \in \mathbf{X}} \min_{i=1\dots n} \|x - X_i\|_2 \not\xrightarrow{P} 0.$$

Доказательство необходимости проводится в 2 шага: на первом покажем, что существует шар $B(c^*, \epsilon)$ для некоторого $c^* \in \mathbf{X}$, в который почти никогда не попадет алгоритм, а на втором построим функцию из множества липшицевых, которая допускает максимум в этом шаре.

Первый шаг. Пусть $\{X_i\}_{i \in \mathbb{N}}$ – последовательность наблюдений, сгенерированная алгоритмом A для функции f . Сначала рассмотрим, что для любого $\epsilon > 0$ ряд $n \in \mathbb{N} \mapsto \mathbb{P}(\sup_{x \in \mathbf{X}} \min_{i=1\dots n} \|x - X_i\|_2 > \epsilon)$ не возрастает и, согласно условию, подразумевается, что

$$\exists \epsilon_1, \epsilon_2 > 0 : \forall n \in \mathbb{N}, \quad \mathbb{P} \left(\sup_{x \in \mathbf{X}} \min_{i=1\dots n} \|x - X_i\|_2 > \epsilon_1 \right) > \epsilon_2. \quad (*)$$

Рассмотрим теперь последовательность $c_1, \dots, c_{N_1}, N_1 = \mathcal{N}_{\epsilon_1}(\mathbf{X})$ точек из \mathbf{X} , определенных ϵ_1 -уровнем, и предположим, что

$$\forall c \in \{c_1, \dots, c_{N_1}\}, \exists n_c \in \mathbb{N} : \mathbb{P} \left(\bigcap_{i=1}^{n_c} \{X_i \notin B(c, \epsilon_1) \cup \mathbf{X}\} \right) \leq \frac{\epsilon_2}{2N_1}$$

или если $N_2 = \max_{c \in \{c_1, \dots, c_{N_1}\}} n_c$:

$$\forall c \in \{c_1, \dots, c_{N_1}\}, \mathbb{P} \left(\bigcap_{i=1}^{N_2} \{X_i \notin B(c, \epsilon) \cap \mathbf{X}\} \right) \leq \frac{\epsilon_2}{2N_1}.$$

Однако, так как c_1, \dots, c_{N_1} определяют ϵ_1 -уровень \mathbf{X} , то это означает следующее:

$$\begin{aligned} \mathbb{P} \left(\sup_{x \in \mathbf{X}} \min_{i=1 \dots N_2} \|x - X_i\|_2 \leq \epsilon_1 \right) &\geq \mathbb{P} \left(\bigcap_{j=1}^{N_1} \bigcup_{i=1}^{N_2} \{X_i \in B(c_j, \epsilon_1) \cap \mathbf{X}\} \right) \\ &= 1 - \mathbb{P} \left(\bigcup_{j=1}^{N_1} \bigcap_{i=1}^{N_2} \{X_i \notin B(c_j, \epsilon_1) \cap \mathbf{X}\} \right) \\ &\geq 1 - \sum_{j=1}^{N_1} \mathbb{P} \left(\bigcap_{i=1}^{N_2} \{X_i \notin B(c_j, \epsilon) \cap \mathbf{X}\} \right) \\ &\geq 1 - N_1 \times \frac{\epsilon_2}{2N_1} \\ &= 1 - \frac{\epsilon_2}{2} \end{aligned}$$

что противоречит условию (*). Следовательно, имеем

$$\exists c^* \in \{c_1, \dots, c_{N_\epsilon}\} : \forall n \in \mathbb{N}, \mathbb{P} \left(\bigcap_{i=1}^n \{X_i \notin B(c^*, \epsilon_1) \cap \mathbf{X}\} \right) \geq \frac{\epsilon_2}{2N_1}$$

Второй шаг. Основываясь на центре $c^* \in \mathbf{X}$, можно представить функцию $\tilde{f} : \mathbf{X} \mapsto \mathbb{R}$, определенную для всех $x \in \mathbf{X}$, следующим образом:

$$\tilde{f}(x) = \begin{cases} f^*(x) + 3 \left(1 - \frac{\|c^* - x\|_2}{\epsilon_1} \right) \times \left(\max_{x \in \mathbf{X}} f^*(x) - \min_{x \in \mathbf{X}} f^*(x) \right), & \text{если } x \in B(c^*, \epsilon_1) \\ f^*(x), & \text{иначе,} \end{cases}$$

которая является липшицевой и имеет максимум в шаре $B(c^*, \epsilon_1)$. Однако, так как \tilde{f} и f^* могут не отличаться в области $\mathbf{X}/B(c^*, \epsilon_1)$, для $n \in \mathbb{N}$ имеем:

$$\begin{aligned} \mathbb{P} \left(\max_{x \in \mathbf{X}} \tilde{f}(x) - \max_{i=1 \dots n} \tilde{f}(X'_i) > \max_{x \in \mathbf{X}} f^*(x) \right) &\geq \mathbb{P} \left(\bigcap_{i=1}^n \{X'_i \notin B(c, \epsilon_2) \cap \mathbf{X}\} \right) \\ &= \mathbb{P} \left(\bigcap_{i=1}^n \{X_i \notin B(c, \epsilon_2) \cap \mathbf{X}\} \right) \\ &\geq \frac{\epsilon_2}{2N_1} > 0, \end{aligned}$$

где X'_1, \dots, X'_n обозначает последовательность точек вычислений, сгенерированная алгоритмом A для функции \tilde{f} .

Таким образом, данный вывод показывает, что существует такая функция $\tilde{f} \in \bigcup_{k \geq 0} Lip(k)$, что для нее не выполняется сходимость:

$$\max_{i=1 \dots n} \tilde{f}(X_i) \not\xrightarrow{p} \max_{x \in \mathbf{X}} \tilde{f}(x).$$

Однако это противоречит определению 1.2 согласованности алгоритма на множестве липшицевых функций. Следовательно, условие

$$\sup_{x \in \mathbf{X}} \min_{i=1 \dots n} \|X_i - x\|_2 \xrightarrow{p} 0, \forall f \in \bigcup_{k \geq 0} Lip(k)$$

является необходимым. \square

Далее, исследуем наилучшую производительность, которая может быть достигнута любым алгоритмом с конечным числом вычислений функции. Начнем с утверждения о том, что любой алгоритм может в любое время понести сколь угодно большие потери по классу липшицевых функций.

Утверждение 1.2. *Рассмотрим алгоритм глобальной оптимизации A . Для произвольной константы $C > 0$, любого $n \in \mathbb{N}$ и $\delta \in (0, 1)$, существует функция $\tilde{f} \in \bigcup_{k \geq 0} Lip(k)$, зависящая только от (A, C, n, δ) , для которой с вероятностью не меньшей $1 - \delta$ имеем*

$$C \leq \max_{x \in \mathbf{X}} \tilde{f}(x) - \max_{i=1 \dots n} \tilde{f}(X_i).$$

Доказательство. Данное доказательство основано на аргументах доказательства Теоремы 1 из [49]. Выберем любой алгоритм $A \in \mathcal{A}$ и константу $C > 0$, зафиксируем $n \in \mathbb{N}$ и $\delta \in (0, 1)$ и установим $n_\delta = \lceil (n/\delta)^{1/d} \rceil$. По определению радиуса $rad(\mathbf{X})$ знаем, что существует некоторая точка $x \in \mathbf{X}$ такая, что $x + [0, 2rad(\mathbf{X})/\sqrt{d}]^d \subseteq \mathbf{X}$. Тогда можно определить для всех $I \in \{1, \dots, n_\delta\}^d$ центры c_I гиперкубов H_I , у которых стороны равны $D = 2rad(\mathbf{X})/(\sqrt{d}N_\delta)$ и которые покрывают \mathbf{X} , то есть $\bigcup_I H_I = x + [0, 2rad(\mathbf{X})/\sqrt{d}]^d \subseteq \mathbf{X}$. Теперь пусть X_1, \dots, X_n есть последовательность n точек вычислений, сгенерированная алгоритмом A для константной функции $f_0 : x \in \mathbf{X} \mapsto 0$ и определим для всех I событие

$$E_I = \bigcap_{i=1}^n \{X_i \notin Int(H_I)\}.$$

Так как внутренние части N_δ^d гиперкубов не пересекаются, то из этого следует

$$N_\delta^d \times \max_I \mathbb{P}(E_I) \geq \sum_I \mathbb{P}(E_I) = \mathbb{E} \left[\sum_I \mathbb{I}\{E_I\} \right] \geq N_\delta^d - n$$

Следовательно, существует I^* , зависящее только от A , которое максимизирует вероятность выше:

$$\mathbb{P}(E_{I^*}) \geq \frac{N_\delta^d - n}{N_\delta^d} = 1 - \frac{n}{\lceil (n/\delta)^{1/d} \rceil^d} \geq 1 - \delta$$

Теперь, используя центр c_{I^*} гиперкуба H_{I^*} , можем представить функцию $\tilde{f} \in \bigcup_{k \geq 0} Lip(k)$, определенную для всех $x \in \mathbf{X}$, следующим образом

$$\tilde{f}(x) = \begin{cases} C \times (1 - 2\|c_{I^*} - x\|_2/D), & \text{если } \|c_{I^*} - x\|_2 \leq D/2 \\ 0, & \text{иначе.} \end{cases}$$

Однако, так как функции \tilde{f} и f_0 могут различаться в пространстве \mathbf{X}/H_{I^*} , тогда имеем:

$$\mathbb{P}\left(\max_{x \in \mathbf{X}} \tilde{f}(x) - \max_{i=1 \dots n} \tilde{f}(X'_i) \geq C\right) \geq \mathbb{P}\left(\bigcap_{i=1}^n \{X'_i \notin \text{Int}(H_{I^*})\}\right) = \mathbb{P}(E_{I^*}) \geq 1 - \delta,$$

где X'_1, \dots, X'_n обозначает последовательность точек вычисления, сгенерированная алгоритмом A для функции \tilde{f} . Полученное выражение доказывает справедливость изначального утверждения. \square

Этот результат может быть не слишком удивительным, поскольку класс липшицевых функций включает функции с конечными, но сколь угодно большими вариациями. При рассмотрении подкласса функций с фиксированной константой Липшица появляется возможность получить оценки за конечное время на минимаксную скорость сходимости.

Следующее утверждение показывает сходимость метода случайного поиска, которое будет полезным для дальнейших рассуждений.

Утверждение 1.3. Пусть $\mathbf{X} \subset \mathbb{R}^d$ непустое, компактное и выпуклое множество и пусть $f \in \text{Lip}(k)$ будет k -липшицевой функцией, определенной на множестве \mathbf{X} для некоторого $k \geq 0$. Тогда для любого $n \in \mathbb{N}$ и любой $\delta \in (0, 1)$ имеем с вероятностью по крайней мере $1 - \delta$ следующее неравенство:

$$\max_{x \in \mathbf{X}} f(x) - \max_{i=1 \dots n} f(X_i) \leq k \cdot \text{diam}(\mathbf{X}) \cdot \left(\frac{\ln(1/\delta)}{n}\right)^{(1/d)},$$

где X_1, \dots, X_n обозначают последовательность n независимых случайных величин из равномерного распределения ($X \sim \mathbb{U}(\mathbf{X})$).

Доказательство. Выберем любой $n \in \mathbb{N}$ и любую $\delta \in (0, 1)$, и пусть $\epsilon = k \cdot \text{diam}(\mathbf{X}) \cdot (\ln(1/\delta)/n)^{1/d}$ есть значение верхней границы и $\mathbf{X}_\epsilon = \{x \in \mathbf{X} : f(x) \geq \max_{x \in \mathbf{X}} f(x) - \epsilon\}$ соответствующая линия уровня. Поскольку утверждение тривиально при $n \leq \ln(1/\delta)$, рассмотрим $n > \ln(1/\delta)$. Заметим теперь, что так как $f \in \text{Lip}(k)$, то для любого $x^* \in \arg \max_{x \in \mathbf{X}} f(x)$ имеем $\mathbf{X} \cap B(x^*, \epsilon/k) \subseteq \mathbf{X}_\epsilon$ и $|f(x) - f(x^*)| \leq k \cdot \|x - x^*\|_2 = \epsilon$ для любого $x \in B(x^*, \epsilon/k) \cap \mathbf{X}$. Таким образом, выбрав любой $x^* \in \arg \max_{x \in \mathbf{X}} f(x)$, получаем:

$$\begin{aligned} \mathbb{P}\left(\max_{i=1 \dots n} f(X_i) \geq \max_{x \in \mathbf{X}} f(x) - \epsilon\right) &= \mathbb{P}\left(\bigcup_{i=1}^n \{X_i \in \mathbf{X}_\epsilon\}\right) \\ &= 1 - \mathbb{P}(X_1 \notin \mathbf{X}_\epsilon) \\ &\geq 1 - \mathbb{P}(X_1 \notin \mathbf{X} \cap B(x^*, \epsilon/k))^n \\ &= 1 - \left(1 - \left(\frac{\mu(\mathbf{X} \cap B(x^*, \epsilon/k))}{\mu(\mathbf{X})}\right)^d\right)^n \\ &\geq 1 - \left(1 - \left(\frac{\epsilon}{k \cdot \text{diam}(\mathbf{X})}\right)^d\right)^n \\ &= 1 - \left(1 - \frac{\ln(1/\delta)}{n}\right)^n \\ &\geq 1 - \delta. \end{aligned}$$

\square

Утверждение 1.4 (Адаптировано из [49]). Для любой константы Липшица $k \geq 0$ и любого $n \in \mathbb{N}$ справедливо следующее неравенство:

$$c_1 \cdot k \cdot n^{-\frac{1}{d}} \leq \inf_{A \in \mathcal{A}} \sup_{f \in \text{Lip}(k)} \mathbb{E} \left[\max_{x \in \mathbf{X}} f(x) - \max_{i=1 \dots n} f(X_i) \right] \leq c_2 \cdot k \cdot n^{-\frac{1}{d}},$$

где $c_1 = \text{rad}(\mathbf{X})/(8\sqrt{d})$, $c_2 = \text{diam}(\mathbf{X}) \times d!$ и математическое ожидание принимает последовательность X_1, \dots, X_n точек вычислений, сгенерированная алгоритмом A для функции f .

Доказательство. (Нижняя граница). Возьмем любое число $n \in \mathbb{N}$ и пусть имеется величина $D = 2\text{rad}(\mathbf{X})/(\sqrt{d}\lceil(2n)^{1/d}\rceil)$. Можно легко показать, воспроизведя шаги из Утверждения 1.2 и приняв δ равной $1/2$, что для любого алгоритма глобальной оптимизации A существует функция $\tilde{f}_A \in \text{Lip}(k)$, определенная как

$$\tilde{f}_A(x) = \begin{cases} k\frac{D}{2} - k \cdot \|c_A - x\|_2, & \text{если } \|c_A - x\|_2 \leq D/2 \\ 0, & \text{иначе} \end{cases}$$

для некоторого центра $c_A \in \mathbf{X}$, зависящего только от выбора алгоритма, для которого имеем:

$$\mathbb{P}\left(\max_{x \in \mathbf{X}} \tilde{f}_A(x) - \max_{i=1 \dots n} \tilde{f}_A(X_i) \geq k \cdot \frac{D}{2}\right) \geq \frac{1}{2},$$

где X_1, \dots, X_n есть последовательность точек вычисления, сгенерированная алгоритмом A для функции \tilde{f}_A . Поэтому, используя определение точной верхней границы и неравенство Маркова, имеем, что для любого $A \in \mathcal{A}$

$$\begin{aligned} \sup_{f \in \text{Lip}(k)} \mathbb{E}\left[\max_{x \in \mathbf{X}} f(x) - \max_{i=1 \dots n} f(X_i)\right] &\geq \mathbb{E}\left[\max_{x \in \mathbf{X}} \tilde{f}_A(x) - \max_{i=1 \dots n} \tilde{f}_A(X_i)\right] \\ &\geq \frac{kD}{2} \times \mathbb{P}\left(\max_{x \in \mathbf{X}} \tilde{f}_A(x) - \max_{i=1 \dots n} \tilde{f}_A(X_i) \geq k \cdot \frac{D}{2}\right) \\ &\geq k \cdot \frac{\text{rad}(\mathbf{X})}{8\sqrt{d}} \cdot n^{-\frac{1}{d}}. \end{aligned}$$

Поскольку данное неравенство справедливо для любого алгоритма A , доказательство существования нижней границы завершено.

(Верхняя граница). Последовательно используя следующие факты:

1. точная нижняя граница минимаксных потерь, понесенных всеми алгоритмами, есть оценка верхней границы потерь алгоритма случайного поиска;
2. для любой случайной положительной величины выполняется:

$$\mathbb{E}[X] = \int_{t=0}^{\infty} \mathbb{P}(X \geq t) dt;$$

3. Утверждение 1.3,

а также, введя переменную $u = n(t/\text{diam}(\mathbf{X}))^{1/d}$, получаем:

$$\begin{aligned} \inf_{A \in \mathcal{A}} \sup_{f \in \text{Lip}(k)} \mathbb{E}\left[\max_{x \in \mathbf{X}} f(x) - \max_{i=1 \dots n} f(X_i)\right] &\leq \sup_{f \in \text{Lip}(k)} \mathbb{E}\left[\max_{x \in \mathbf{X}} f(x) - \max_{i=1 \dots n} f(X_i)\right] \\ &\leq \int_0^{\infty} \exp\left\{-n(t/k \cdot \text{diam}(\mathbf{X}))^{1/d}\right\} dt \\ &= k \cdot \text{diam}(\mathbf{X}) \cdot n^{-d} \cdot d \cdot \int_0^{\infty} u^{d-1} e^{-u} du \\ &= k \cdot \text{diam}(\mathbf{X}) \cdot n^{-d} \cdot d \cdot \Gamma(d), \end{aligned}$$

где X'_1, \dots, X'_n обозначают последовательность n независимых наблюдений из нормального распределения, и $\Gamma(\cdot)$ есть гамма-функция. Вспоминая, что $\Gamma(d) = (d-1)!$ для любого $d \in \mathbb{N}$, завершаем доказательство. \square

Данная минимаксная скорость сходимости порядка $\Theta(n^{-1/d})$ все ещё может быть достигнута любым методом с оптимальной скоростью покрытия порядка $O(n^{-1/d})$. Поскольку

$$\mathbb{E} \left[\max_{x \in \mathbf{X}} f(x) - \max_{i=1 \dots n} f(X_i) \right] \leq k \times \mathbb{E} \left[\sup_{x \in \mathbf{X}} \min_{i=1 \dots n} \|x - X_i\|_2 \right]$$

для любой $f \in Lip(k)$, то оптимальная скорость покрытия обязательно повлечет за собой минимаксную эффективность. Однако, как видно из доказательства Утверждения 1.4, функции, построенные для доказательства предельной оценки, представляют собой всплески, которые почти постоянны всюду и не предоставляют большой интерес с практической точки зрения.

Данные теоретические знания позволяют далее спроектировать и проанализировать следующие алгоритмы:

1. Алгоритм с фиксированной константой Липшица $k \geq 0$, который обеспечивает минимаксную эффективность, а также предоставляет экспоненциально убывающую скорость схождения для большого подмножества функций;
2. Алгоритм, который не требует знаний константы Липшица и обеспечивает производительность, сравнимую с алгоритмом, где эта константа известна;
3. Алгоритм с неизвестной константой, но сходящийся значительно быстрее.

2 Оптимизация с фиксированной константой Липшица

В данном разделе рассматривается проблема оптимизации неизвестной функции f , для которой известно $f \in Lip(k)$ и дана константа $k \geq 0$.

2.1 Алгоритм

Входными данными алгоритма *LIPO* [50] являются число n вычислений неизвестной функции, константа Липшица $k \geq 0$, область определения \mathbf{X} и сама неизвестная функция $f \in Lip(k)$. На каждой итерации $t \geq 1$ выбирается случайная величина X_{t+1} из равномерного распределения из области определения \mathbf{X} , и алгоритм решает, вычислять значение функции в этой точке или нет. В самом деле, он вычисляет значение в точке X_{t+1} тогда и только тогда, когда значение верхней границы возможных значений, вычисленное в этой точке и оценено на основе предыдущих значений, по крайней мере, не ниже значения наилучшей наблюдаемой оценки $\max_{i=1\dots t} f(X_i)$, то есть:

$$UB : x \rightarrow \min_{i=1\dots t} f(X_i) + k \cdot \|x - X_i\|_2$$

Псевдокод алгоритма представлен ниже.

Algorithm 1 *LIPO*(n, k, \mathbf{X}, f)

1. Initialization:

Let $X_1 \sim \mathbb{U}(\mathbf{X})$

Evaluate $f(X_1), t \leftarrow 1$

2. Iterations:

while $t < n$ **do**

Let $X_{t+1} \sim \mathbb{U}(\mathbf{X})$

if $\min_{i=1\dots t} (f(X_i) + k \cdot \|X_{t+1} - X_i\|_2) \geq \max_{i=1\dots t} f(X_i)$ **then**

Evaluate $f(X_{t+1}), t \leftarrow t + 1$

end if

end while

3. Output:

Return $X_{\hat{x}_n}$, where $\hat{i}_n \in \arg \max_{i=1\dots n} f(X_i)$

Более формально, механизм, лежащий в основе правила принятия решений, можно объяснить с помощью активного подмножества согласованных функций, которые рассматривались при активном обучении ([51] или [52]).

Определение 2.1. *Активное подмножество k -липшицевых функций, согласующееся с неизвестной функцией f и выборкой $(X_1, f(X_1)), \dots, (X_t, f(X_t))$ для $t \geq 1$, определяется как*

$$\mathcal{F}_{k,t} := \{g \in Lip(k) : \forall i \in \{1 \dots t\}, g(X_i) = f(X_i)\}$$

Действительно, из этого определения можно восстановить подмножество точек, которые могут фактически максимизировать целевую функцию f .

Определение 2.2. Используя те же обозначения из Определения 2.1, обозначим подмножество потенциальных точек максимума, оцененных для любой выборки $t \geq 1$ вычислений с константой $k \geq 0$, следующим образом:

$$\mathbf{X}_{k,t} := \left\{ x \in \mathbf{X} : \exists g \in \mathcal{F}_{k,t} | x \in \arg \max_{x \in \mathbf{X}} g(x) \right\}.$$

Теперь можно установить эквивалентность с правилом принятия решения алгоритма *LIPO*

Лемма 1. Если $\mathbf{X}_{k,t}$ определяет множество потенциальных точек максимума (из Определения 2.2), то имеет место следующее соответствие:

$$x \in \mathbf{X}_{k,t} \Leftrightarrow \min_{i=1\dots t} f(X_i) + k \cdot \|x - X_i\|_2 \geq \max_{i=1\dots t} f(X_i).$$

Доказательство. Доказательство необходимости является следствием Определения 2.2. Замечая, что функция

$$\tilde{f} : x \mapsto \min_{i=1\dots t} (\max_{i=1\dots t} f(X_i), \min_{i=1\dots t} f(X_i) + k \cdot \|x - X_i\|_2)$$

принадлежит $\mathcal{F}_{k,t}$ и что

$$\arg \max_{x \in \mathbf{X}} \tilde{f}(x) = \left\{ x \in \mathbf{X} : \min_{i=1\dots t} f(X_i) + k \cdot \|x - X_i\|_2 \geq \max_{i=1\dots t} f(X_i) \right\},$$

получаем доказательство достаточности. \square

Из Леммы 1 следует, что алгоритм оценивает функцию только в точках, которые ещё могут быть точками максимума неизвестной функции.

Замечание 1. Вполне вероятно, что алгоритм *LIPO* может быть распространён на условия с зашумлёнными значениями функции, адаптировав идеи, развитые в [51] и [52]. Точнее, когда рассматривается выборка $(X_1, Y_1), \dots, (X_n, Y_n)$, $n \geq 1$ зашумлённых значений, где $Y_i = fX_i + \sigma\epsilon_i$ и $\epsilon_i \sim \mathcal{N}(0, 1)$, у которых эмпирическая среднеквадратичная ошибка

$$R_n(f) := \frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i)^2 = \frac{\sigma^2}{n} \sum_{i=1}^n \epsilon_i^2,$$

вычисленная для f , распределена по закону χ^2 , тогда существует возможный подход, состоящий в использовании ослабленной версии активного подмножества из Определения 2.1

$$\mathcal{F}_{k,\delta,t} := \left\{ g \in \text{Lip}(k) : R_n(g) \leq \frac{\sigma^2}{n} \cdot \chi_{1-\delta,n}^2 \right\},$$

где $\chi_{1-\delta,n}^2$ обозначает распределение χ^2 с квантилем $1 - \delta$ и с n степенями свободы.

Замечание 2. Важно отметить, что предложенная схема оптимизации может быть легко расширена на большее число классов глобально и локально гладких функций путём небольшого изменения правила принятия решения. Например, пусть

$$\mathcal{F}_l = \{ f : \mathbf{X} \rightarrow \mathbb{R} | x^* \text{ уникально и } \forall x \in \mathbf{X}, f(x^*) - f(x) \leq l(x^*, x) \}$$

определяет множество функций, которые являются локально гладкими около их максимумов по отношению к данной метрике $l : \mathbf{X} \times \mathbf{X} \rightarrow \mathbb{R}^+$, тогда из прямого вывода Леммы 1 следует, что правило принятия решений в точке X_{t+1} будет просто заключаться в проверке условия

$$\max_{i=1\dots t} f(X_i) \leq \min_{i=1\dots t} f(X_i) + l(X_{t+1}, X_i).$$

Однако, поскольку целью данной работы является построение быстрого алгоритма для липшицевых функций, приведём результаты сходимости только для указанного варианта алгоритма.

2.2 Анализ сходимости

Начнём с приведения свойства согласованности алгоритма

Утверждение 2.1. *Рассмотрим алгоритм LIPO с фиксированной константой $k \geq 0$. Для $f \in Lip(k)$ и $n \in \mathbb{N}$ имеем, что $\forall y \in \mathbb{R}$,*

$$\mathbb{P}\left(\max_{i=1\dots n} f(X_i) \geq y\right) \geq \mathbb{P}\left(\max_{i=1\dots n} f(X'_i) \geq y\right),$$

где X_1, \dots, X_n есть последовательность n точек, сгенерированных алгоритмом LIPO и X'_1, \dots, X'_n – последовательность из n независимых, равномерно распределённых величин из \mathbf{X} .

Доказательство данного утверждения приводится в [53]. Основываясь на данном результате, можно легко получить за конечное время первую оценку разности между значением истинного максимума и его аппроксимацией.

Утверждение 2.2. *Для любой константы Липшица $k \geq 0$, алгоритм LIPO с параметром k согласуется на множестве k -липшицевых функций, то есть*

$$\forall f \in Lip(k), \max_{i=1\dots n} f(X_i) \xrightarrow{P} \max_{x \in \mathbf{X}} f(x).$$

Доказательство. Для любой функции $f \in Lip(k)$ выберем $n \in \mathbb{N}$, $\epsilon > 0$ и пусть

$$\mathbf{X}_\epsilon = \left\{ x \in \mathbf{X} : f(x) \geq \max_{x \in \mathbf{X}} f(x) - \epsilon \right\}$$

соответствующая линия уровня. Обозначая X'_1, \dots, X'_n последовательность n случайных равномерно распределённых величин из \mathbf{X} и замечая, что $\mu(\mathbf{X}_\epsilon) > 0$, получаем из Утверждения 2.1:

$$\begin{aligned} \mathbb{P}\left(\max_{x \in \mathbf{X}} f(x) - \max_{i=1\dots n} f(X_i) > \epsilon\right) &\leq \mathbb{P}\left(\max_{x \in \mathbf{X}} f(x) - \max_{i=1\dots n} f(X'_i) > \epsilon\right) \\ &= \mathbb{P}\left(\bigcap_{i=1}^n \{X'_i \notin \mathbf{X}_\epsilon\}\right) \\ &\leq \left(1 - \frac{\mu(\mathbf{X}_\epsilon)}{\mu(\mathbf{X})}\right)^n \xrightarrow{n \rightarrow \infty} 0. \end{aligned}$$

□

Следующие результаты показывают, что значение наивысшей оценки, наблюдаемой алгоритмом, всегда не ниже значения случайного поиска.

Следствие 1 (Верхняя граница). *Для любой функции $f \in Lip(k)$, любого $n \in \mathbb{N}$ и $\delta \in (0, 1)$ имеем верхнюю границу (по крайней мере, с вероятностью $1 - \delta$):*

$$\max_{x \in \mathbf{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq k \cdot \text{diam}(\mathbf{X}) \cdot \left(\frac{\ln(1/\delta)}{n}\right)^{1/d}.$$

Данное следствие является обобщением из Утверждений 2.1 и 1.3. Это верхняя граница, показывающая минимаксную оптимальность данного алгоритма, однако не показывает какого-либо улучшения по сравнению со случайным поиском, и её нельзя значительно улучшить без каких-либо дополнительных предположений, как показано ниже.

Утверждение 2.3 (Нижняя граница). Для любого $n \in \mathbb{N}$ и $\delta \in (0, 1)$ существует функция $\tilde{f} \in Lip(k)$, зависящая только от n и δ , для которой с вероятностью $1 - \delta$ имеем:

$$k \cdot rad(\mathbf{X}) \cdot \left(\frac{\delta}{n}\right)^{\frac{1}{d}} \leq \max_{x \in \mathbf{X}} \tilde{f}(x) - \max_{i=1 \dots n} \tilde{f}(X_i).$$

Доказательство. Положим $\delta \in (0, 1)$, $n \in \mathbb{N}$ и пусть $r_{\delta, n} = rad(\mathbf{X})(\delta/n)^{1/d}$ будет значение нижней границы, деленное на k . Так как $rad(\mathbf{X}) > 0$, то обязательно существует точка $x^* \in \mathbf{X}$ такая, что $B(x^*, rad(\mathbf{X})) \subseteq \mathbf{X}$. На основе этой точки, можно представить функцию $\tilde{f} \in Lip(k)$, определенную для всех $x \in \mathbf{X}$, следующим образом:

$$\tilde{f} = \begin{cases} k \cdot r_{\delta, n} - k \cdot \|x - x^*\|_2, & \text{если } x \in B(x^*, r_{\delta, n}) \\ 0, & \text{иначе.} \end{cases}$$

Обозначая теперь $X_1, \dots, X_n \sim \mathcal{U}(\mathbf{X})$ последовательность n точек вычисления, сгенерированная алгоритмом *LIFO* с параметром k для функции \tilde{f} , имеем

$$\begin{aligned} \mathbb{P}\left(\max_{x \in \mathbf{X}} \tilde{f}(x) - \max_{i=1 \dots n} \tilde{f}(X_i) \geq k \cdot r_{\delta, n}\right) &\geq \mathbb{P}\left(\bigcap_{i=1}^n \{X_i \notin B(x^*, r_{\delta, n})\}\right) \\ &= [\mathbb{P}(X_1 \notin B(x^*, r_{\delta, n})) \times \\ &\quad \prod_{i=1}^{n-1} \mathbb{P}\left(X_{i+1} \notin B(x^*, r_{\delta, n}) \mid \bigcap_{t=1}^i \{X_t \notin B(x^*, r_{\delta, n})\}\right)] \\ &= \left(1 - \frac{\mu(B(x^*, r_{\delta, n})) \cap \mathbf{X}}{\mu(\mathbf{X})}\right)^n \\ &\geq \left(1 - \left(\frac{r_{\delta, n}}{rad(\mathbf{X})}\right)^d\right)^n \\ &= \left(1 - \frac{\delta}{n}\right)^n \\ &\geq 1 - \delta \end{aligned}$$

□

Таким образом, в данном разделе был описан алгоритм *LIFO*, а также установлена сходимость этого алгоритма.

3 Оптимизация с неизвестной константой Липшица

В данном разделе рассматривается проблема оптимизации любой неизвестной функции f в классе

$$\bigcup_{k \geq 0} Lip(k)$$

3.1 Алгоритм

Алгоритм *AdaLIPO* [50] является расширением *LIPO*, которое включает оценку константы Липшица и принимает в качестве входных данных параметр $p \in (0, 1)$ и неубывающую последовательность констант Липшица $k_{i \in \mathbb{Z}}$, которая определяет сетку \mathbb{R}^+ (то есть $\forall x > 0, \exists i \in \mathbb{Z} : k_i \leq x \leq k_{i+1}$). Алгоритм инициализируется константой Липшица \hat{k}_1 равной 0 и случайным образом чередуется между двумя отдельными фазами: поиск и эксплуатация. Так, на шаге $t < n$ выбирается случайная величина из распределения Бернулли B_{t+1} с параметром p , которая определяет фазу алгоритма. Если $B_{t+1} = 1$, то алгоритм исследует пространство, оценивая функцию в случайной точке из равномерного распределения. В противном случае, если $B_{t+1} = 0$, то алгоритм использует предыдущие оценки, выполняя итерацию алгоритма *LIPO* с наименьшей константой Липшица последовательности \hat{k}_t , связанной с подмножеством липшицевых функций, которое, вероятно, содержит f . После выполнения итерации оценка константы Липшица обновляется.

Algorithm 2 *AdaLIPO*($n, p, k_{i \in \mathbb{Z}}, \mathbf{X}, f$)

1. Initialization:

Let $X_1 \sim U(\mathbf{X})$

Evaluate $f(X_1), t \leftarrow 1, \hat{k}_1 \leftarrow 0$

2. Iterations:

while $t < n$ **do**

Let $B_{t+1} \sim \mathbb{B}(p)$

if $B_{t+1} = 1$ **then** (Exploration)

Let $X_{t+1} \sim U(\mathbf{X})$

else if $B_{t+1} = 0$ **then** (Exploitation)

Let $X_{t+1} \sim U(\mathbf{X}_{\hat{k}_t, t})$

end if

Evaluate $f(X_{t+1}), t \leftarrow t + 1$

Let $\hat{k}_t := \inf \left\{ k_{i \in \mathbb{Z}} : \max_{i \neq j} \frac{|f(X_i) - f(X_j)|}{\|X_i - X_j\|_2} \right\} \leq k_i$

end while

3. Output:

Return $X_{\hat{x}_n}$, where $\hat{i}_n \in \arg \max_{i=1 \dots n} f(X_i)$

Замечание 3. Можно рассматривать несколько последовательностей констант Липшица различной формы, таких как $k_i = |i|^{\text{sgn}(i)}$, $\ln(1 + |i|^{\text{sgn}(i)})$ или $(1 + \alpha)^i$ для некоторого $\alpha > 0$. Следует отметить, что вычисление оценки с данными последовательностями не вызывает трудностей. Например, для $k_i = (1 + \alpha)^i$ имеем $\hat{k}_t = (a + \alpha)^{i_t}$, где

$$i_t = \left\lceil \frac{\ln(\max_{i \neq j} |f(X_j) - f(X_i)| / \|X_j - X_i\|_2)}{\ln(1 + \alpha)} \right\rceil.$$

Замечание 4. Из-за универсальности алгоритма любая оценка константы Липшица, предложенная в [54] или [55], также может рассматриваться для реализации алгоритма. Однако, поскольку для анализа требуется, чтобы оценка была непротиворечивой, будем рассматривать только ту предложенную оценку, которая обладает таким свойством.

3.2 Анализ сходимости

Прежде чем приступить к анализу алгоритма, мы сначала обеспечим контроль над оценкой константы Липшица на основе выборки случайных оценок, которые будут полезны для анализа его производительности. Точнее, следующий результат иллюстрирует цель использования дискретизации константы Липшица вместо грубой оценки максимального наклона, показывая, что с учетом этой оценки небольшое подмножество функций, содержащих неизвестную функцию, может быть найдено за конечное время.

Утверждение 3.1. Пусть $f \in \bigcup_{k \geq 0} \text{Lip}(k)$ – любая непостоянная Липшицева функция. Тогда, если \hat{k}_t есть оценка константы Липшица в алгоритме AdaLIPPO, вычисленная любой возрастающей последовательностью $k_{i \in \mathbb{Z}} \in \mathbb{R}^+$ с выборкой $(X_1, f(X_1)), \dots, (X_t, f(X_t))$ для $t \geq 2$ вычислений, где X_1, \dots, X_t – случайные равномерно распределённые величины из \mathbf{X} , имеем:

$$\mathbb{P}(f \in \text{Lip}(\hat{k}_t)) \geq 1 - (1 - \Gamma(f, k_{i^*-1}))^{\lfloor t/2 \rfloor},$$

где

$$\Gamma(f, k_{i^*-1}) := \mathbb{P}\left(\frac{|f(X_1) - f(X_2)|}{\|X_1 - X_2\|_2} > k_{i^*-1}\right) > 0,$$

$$i^* = \min\{i \in \mathbb{Z} : f \in \text{Lip}(k_i)\}.$$

Доказательство. Для $t \geq 2$ рассмотрим любую непостоянную функцию $f \in \bigcup_{k \geq 0} \text{Lip}(k)$ и пусть $i^* = \min\{i \in \mathbb{Z} : f \in \text{Lip}(k_i)\}$. Для доказательства декоррелируем выборку и используем тот факт, что $(X_1, X_{\lfloor t/2 \rfloor + 1}), \dots, (X_{\lfloor t/2 \rfloor}, X_{2\lfloor t/2 \rfloor})$ формирует последовательность $\lfloor t/2 \rfloor$ независимых и одинаково распределённых пар величин $(X, X') \sim \mathcal{U}(\mathbf{X} \times \mathbf{X})$:

$$\begin{aligned} \mathbb{P}(f \in \text{Lip}(\hat{k}_t)) &= \mathbb{P}\left(\bigcup_{i \neq j}^t \{|f(X_i) - f(X_j)| > k_{i^*-1} \cdot \|X_i - X_j\|_2\}\right) \\ &\geq \mathbb{P}\left(\bigcup_{i=1}^{\lfloor t/2 \rfloor} \{|f(X_i) - f(X_{\lfloor t/2 \rfloor + i})| > k_{i^*-1} \cdot \|X_i - X_{\lfloor t/2 \rfloor + i}\|_2\}\right) \\ &= 1 - \mathbb{P}\left(\frac{|f(X_1) - f(X_2)|}{\|X_1 - X_2\|_2} \leq k_{i^*-1}\right)^{\lfloor t/2 \rfloor} \\ &= 1 - (1 - \Gamma(f, k_{i^*-1}))^{\lfloor t/2 \rfloor}. \end{aligned}$$

Осталось показать, что $\Gamma(f, k_{i^*-1}) > 0$. Заметим, что $f \in \text{Lip}(k_{i^*})$, и функция $F : (x, x') \mapsto |f(x) - f(x')| - k_{i^*-1} \cdot \|x - x'\|_2$ является непрерывной. Однако, так как $f \notin \text{Lip}(k_{i^*-1})$, мы знаем, что существует некоторая пара $(x_1, x_2) \in \mathbf{X} \times \mathbf{X}$ такая, что $F(x_1, x_2) > 0$. Следовательно, из непрерывности функции F обязательно существует некоторый $\epsilon > 0$ такой, что:

$$\forall (x, x') \in B(x_1, \epsilon) \cap \mathbf{X} \times B(x_2, \epsilon) \cap \mathbf{X}, \quad F(x, x') > 0.$$

□

Данные результаты позволяют нам проанализировать алгоритм *AdaLIPO*. Этот метод во многом повторяет выводы предыдущего (может, за исключением точности границ сходимости), так как использует схожие идею и реализацию. Так, доказательство согласованности алгоритма *AdaLIPO* повторяет доказательство Утверждения 1.1 для общего случая. Отличием будет являться верхняя граница сходимости.

Утверждение 3.2 (Верхняя граница). *Рассмотрим алгоритм AdaLIPO с параметром $p \in (0, 1)$ и последовательностью констант Липшица k , которая определяет сетку \mathbb{R}^+ . Для любой непостоянной функции $f \in \bigcup_{k \geq 0} Lip(k)$, любого $n \in \mathbb{N}$ и $\delta \in (0, 1)$, имеем, по крайней мере, с вероятностью $1 - \delta$ следующую оценку:*

$$\max_{x \in \mathbf{X}} f(x) - \max f(X_i) \leq k_{i^*} \cdot \text{diam}(\mathbf{X}) \cdot \left(\frac{5}{p} + \frac{2 \ln(\delta/3)}{p \ln(1 - \Gamma(f, k_{i^*-1}))} \right)^{1/d} \cdot \left(\frac{\ln(3/\delta)}{n} \right)^{1/d},$$

где $\Gamma(f, k_{i^*-1})$ и i^* определены в Утверждении 3.1.

Доказательство. Пусть $\delta \in (0, 1)$ и примем

$$N_1 = 2 + \left\lceil 2 \frac{\ln(\delta/3)}{\ln(1 - \Gamma(f, k_{i^*-1}))} \right\rceil,$$

$$N_2 = \left\lceil \left(\frac{\sqrt{\ln(3/\delta)/2} + 4N_1 p - \sqrt{\ln(3/\delta)/2}}{2p} \right)^2 \right\rceil.$$

Для любого $n > N_2$ докажем теорему в три шага:

Первый шаг. При известных N_1 и N_2 неравенство Хёфдинга гарантирует, что

$$\mathbb{P} \left(\sum_{i=1}^{N_2} B_i > N_1 \right) \geq 1 - \frac{\delta}{3}.$$

То есть, мы знаем, что после N_2 итераций с вероятностью $1 - \delta/3$ получим N_1 случайных и равномерно распределенных точек вычисления из \mathbf{X} .

Второй шаг. Используя Утверждение 3.1 и первые N_1 точек вычисления, которые выбраны независимо и равномерно из \mathbf{X} , получаем, что после N_2 итераций и события $\left\{ \sum_{i=1}^{N_2} B_i \geq N_1 \right\}$ константа Липшица k_{i^*} оценивается с вероятностью $1 - \delta/3$, то есть

$$\mathbb{P} \left(\forall t \geq N_2 + 1, \hat{k}_t = k_{i^*} \mid \sum_{i=1}^{N_2} B_i \geq N_1 \right) \geq 1 - \frac{\delta}{3}$$

Третий шаг. Наконец, так как оценка константы Липшица \hat{k}_t удовлетворяет $f \in Lip(\hat{k}_t)$ для всех $t \geq N_2 + 1$ в событии выше, можно легко показать, воспроизведя шаги доказательства Утверждения 2.1, что в зависимости от события $\left\{ \forall t \geq N_2 + 1, \hat{k}_t = k_{i^*} \right\} \cap \left\{ \sum_{i=1}^{N_2} B_i \geq N_1 \right\}$ алгоритм всегда будет быстрее, чем случайный поиск с $n - N_2$ независимых значений $X' \sim \mathbb{U}(\mathbf{X})$. Поэтому, используя Утверждение 1.3 и следующие элементарные неравенства:

$$\lceil x \rceil \leq x + 1;$$

$$\lfloor x \rfloor \geq x - 1;$$

$$\sqrt{x+y} - \sqrt{x} \leq \sqrt{y},$$

а также учитывая, что $N_2 < n$, мы получаем с вероятностью по меньшей мере $(1 - \delta/3)^3$:

$$\begin{aligned}
\max_{x \in \mathbf{X}} f(x) - \max_{i=1 \dots n} f(X_i) &\leq k_{i^*} \cdot \text{diam}(\mathbf{X}) \cdot \left(\frac{\ln(3/\delta)}{n - N_2} \right)^{1/d} \\
&= k_{i^*} \cdot \text{diam}(\mathbf{X}) \cdot \left(\frac{n}{n - N_2} \right)^{1/d} \cdot \left(\frac{\ln(3/\delta)}{n} \right)^{1/d} \\
&\leq k_{i^*} \cdot \text{diam}(\mathbf{X}) \cdot (1 + N_2)^{1/d} \cdot \left(\frac{\ln(3/\delta)}{n} \right)^{1/d} \\
&\leq k_{i^*} \cdot \text{diam}(\mathbf{X}) \cdot \left(\frac{5}{p} + \frac{2 \ln(\delta/3)}{p \cdot \ln(1 - \Gamma(f, k_{i^* - 1}))} \right)^{1/d} \cdot \left(\frac{\ln(3/\delta)}{n} \right)^{1/d}.
\end{aligned}$$

Этот результат также распространяется на случай, когда $n \leq N_2$, отметив, что граница превышает $k_{i^*} \cdot \text{diam}(\mathbf{X})$ и, следовательно, доказательство тривиально. \square

Таким образом, был произведен анализ сходимости алгоритма *AdaLIPO*.

4 Модификация метода оптимизации с неизвестной константой Липшица

4.1 Алгоритм

Алгоритм имеет схожую идею с *AdaLIPO*, однако вместо максимально возможной оценки используется вероятность улучшения значения неизвестной функции. Данный подход вдохновлен одной из функций выбора в Байесовской оптимизации: *Probability of Improvement (poi)*. В самом деле, на шаге $t < n$ метод вычисляет значение в точке с наибольшей вероятностью улучшения poi :

$$poi = \frac{f_{min} - \max_{i=1\dots t} f(X_i)}{f_{max} - f_{min}},$$

где

$$f_{min} = \min_{i=1\dots t} (f(X_i) + k_t \cdot \|X_{t+1} - X_i\|_2),$$

$$f_{max} = \max_{i=1\dots t} (f(X_i) - k_t \cdot \|X_{t+1} - X_i\|_2).$$

Algorithm 3 `mod_adaLIPO` ($n, p, k_{i \in \mathbb{Z}}, \mathbf{X}, f$)

1. Initialization:

Let $X_1 \sim U(\mathbf{X})$

Evaluate $f(X_1), t \leftarrow 1, \hat{k}_1 \leftarrow 0$

2. Iterations:

while $t < n$ **do**

Let $X_{t+1} \sim \mathbb{U}(\mathbf{X}), poi_{max} \leftarrow 0$

Evaluate $f_{min} = \min_{i=1\dots t} (f(X_i) + \hat{k}_t \cdot \|X_{t+1} - X_i\|_2)$

Evaluate $f_{max} = \max_{i=1\dots t} (f(X_i) - \hat{k}_t \cdot \|X_{t+1} - X_i\|_2)$

if $f_{min} > \max_{i=1\dots t} f(X_i)$ **then**

$$poi = \frac{f_{min} - \max_{i=1\dots t} f(X_i)}{f_{max} - f_{min}}$$

end if

if $poi > poi_{max}$ **then**

$$poi_{max} = poi$$

Evaluate $f(X_{t+1}), t \leftarrow t + 1$

end if

Let $\hat{k}_t := \inf \left\{ k_{i \in \mathbb{Z}} : \max_{i \neq j} \frac{|f(X_i) - f(X_j)|}{\|X_i - X_j\|_2} \right\} \leq k_i$

end while

3. Output:

Return $X_{\hat{x}_n}$, where $\hat{i}_n \in \arg \max_{i=1\dots n} f(X_i)$

Данный алгоритм имеет схожую структуру с алгоритмами *LIPO* и *AdaLIPO*, поэтому на основе Утверждений 1.3 и 1.1 алгоритм является согласованным. Поиск границы сходимости алгоритма не тривиален, поэтому данный вопрос остается на будущее изучение метода.

5 Результаты

Здесь представлено сравнение модифицированного алгоритма *AdaLIPO* (*mod_adaLIPO*) с его исходной версией (*adaLIPO*), алгоритмом *LIPO* (*LIPO*), а также со следующими известными методами в области решения задач глобальной оптимизации:

- Байесовская оптимизация [39] с различными функциями выбора:
 - «Ожидаемое улучшение» (*Expected Improvement*) – **BO_ei**;
 - «Вероятность улучшения» (*Probability of Improvement*) – **BO_poi**;
 - «Верхняя граница доверительного интервала» (*Upper Confidence Bound*) – **BO_ucb**.

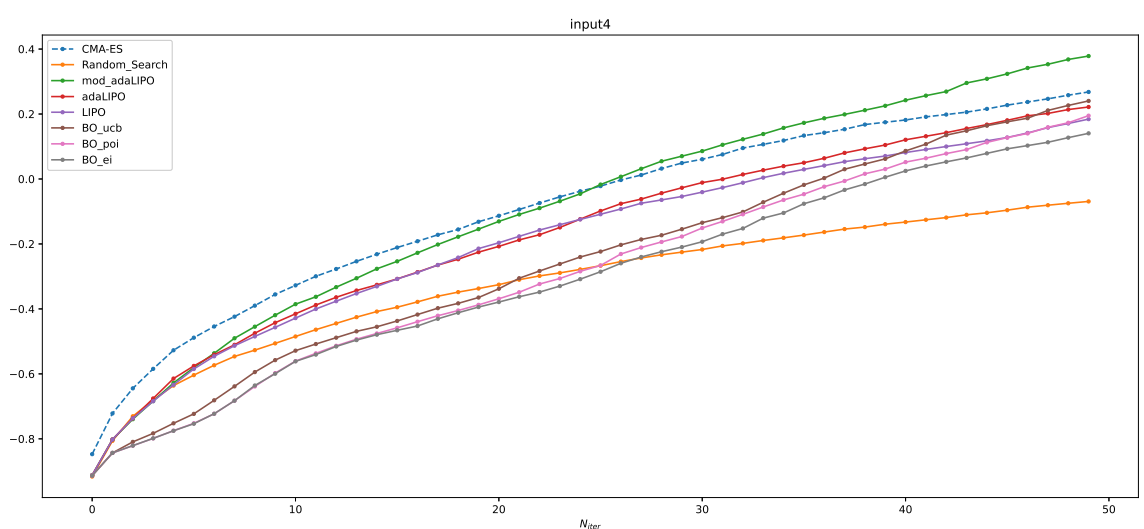
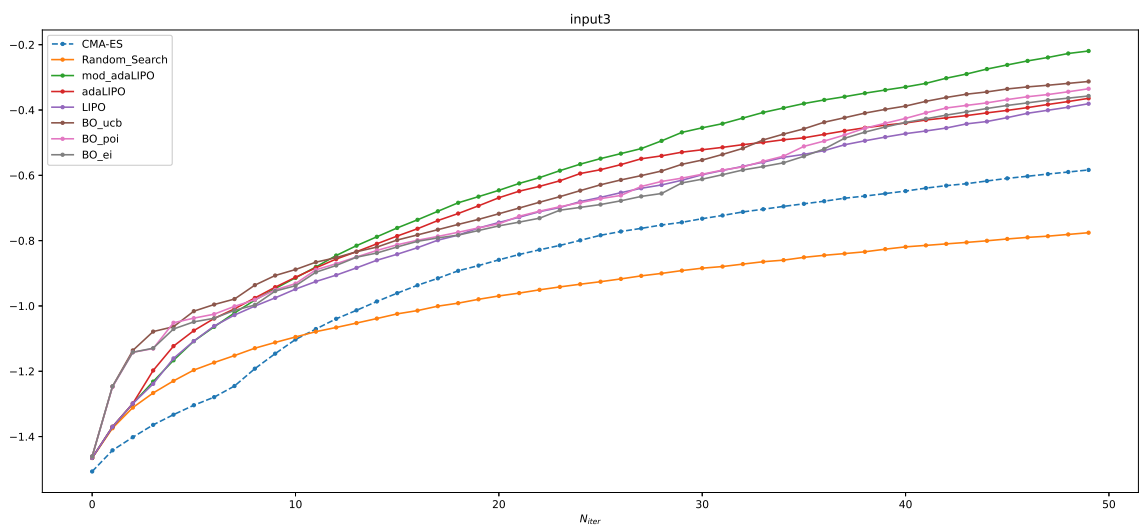
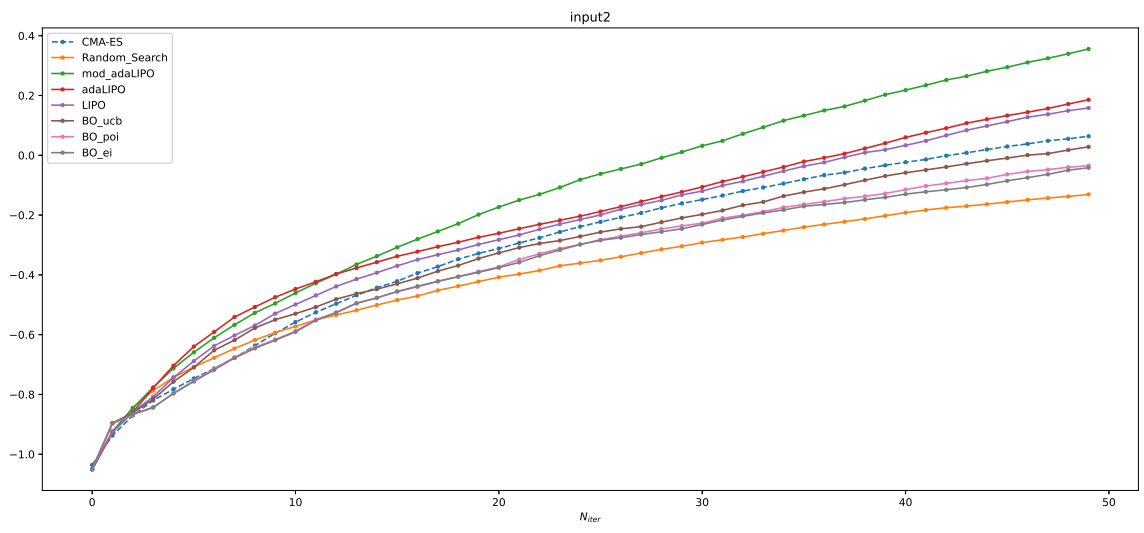
Данный метод использует распределение по функциям с различными параметрами для построения суррогатной модели неизвестной функции. Параметры распределения оцениваются в процессе оптимизации.

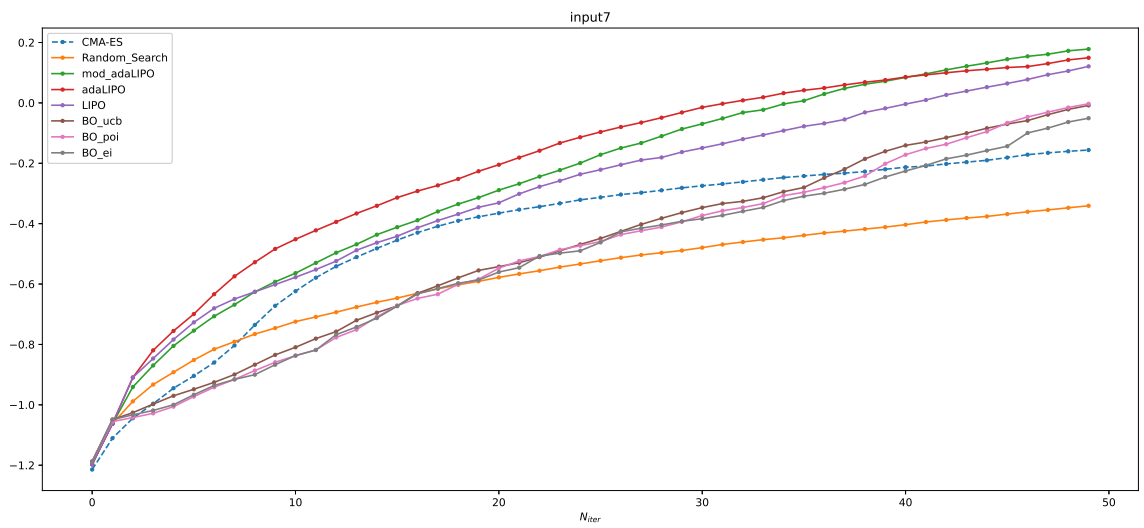
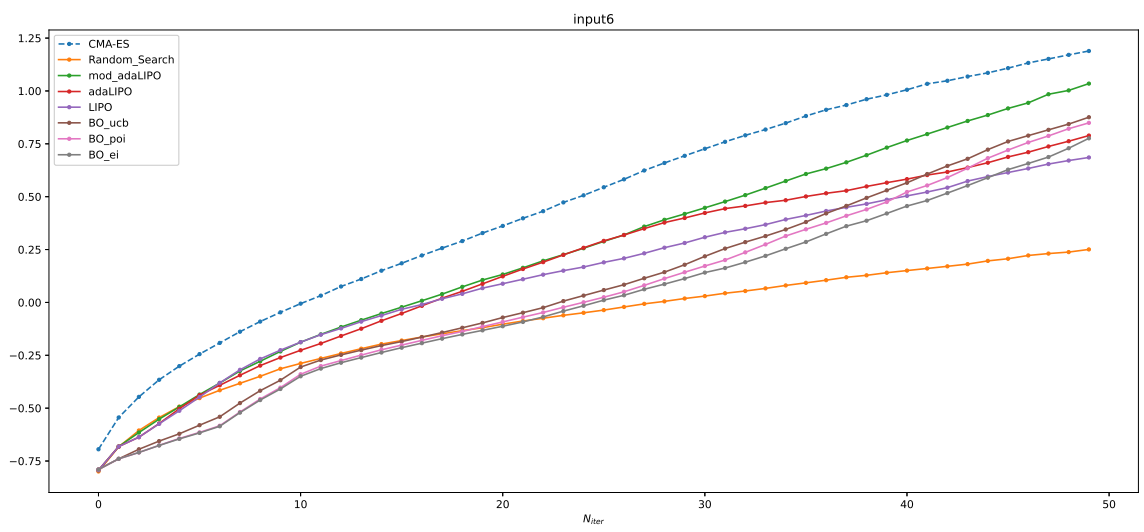
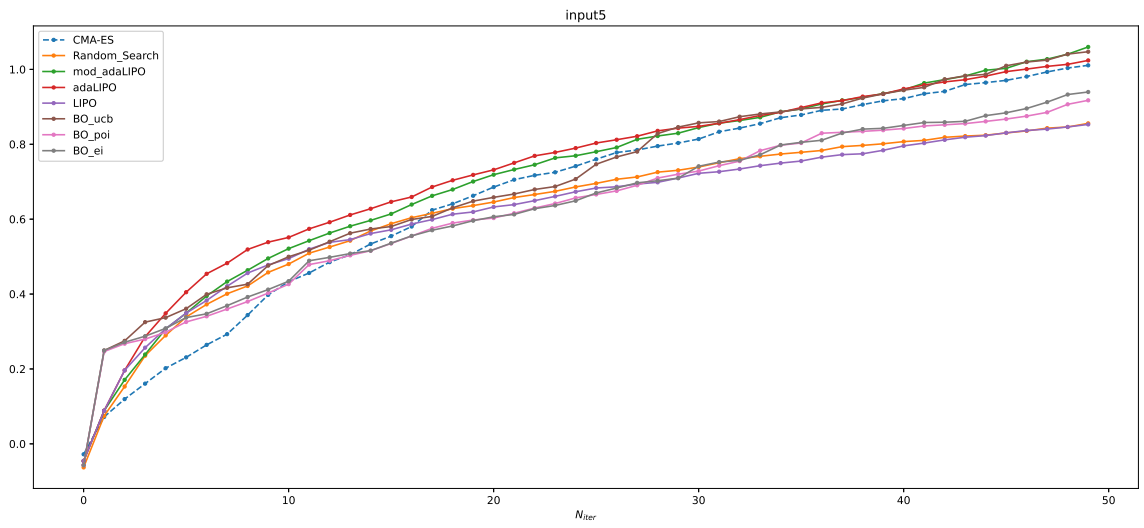
- Эволюционный алгоритм *CMA-ES* [56]. Он выбирает следующие точки вычисления в соответствии с многомерным нормальным распределением со средним вектором и ковариационной матрицей, вычисленными на основе предыдущих значений – **CMA-ES**.
- Случайный поиск, который представляет случайное покрытие области поиска точками из равномерного распределения – **Random_Search**.

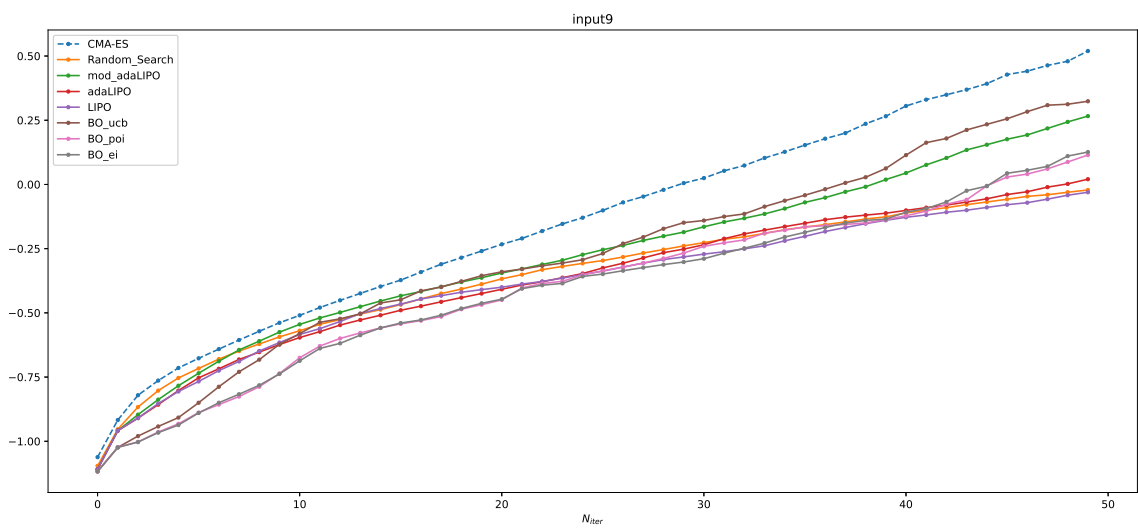
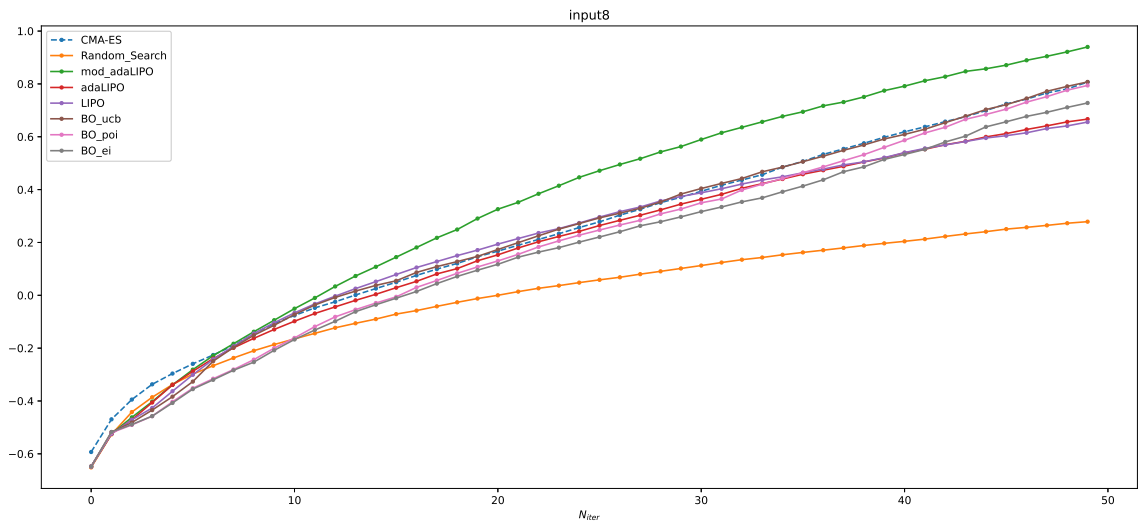
Данные для сравнения эффективности алгоритмов были получены из реальной задачи покрытия базовых станций сотовой связи. Таким образом, имеется 8 различных датасетов (*input2-8*), в каждом из которых имеется следующее число задач:

input2	input3	input4	input5	input6	input7	input8	input9	Σ
4688	5000	5000	4841	4629	4705	4398	3769	37030

Каждый алгоритм запускался на каждом датасете для каждой задачи 51 раз и выполнял 50 итераций, получая невязки ($F_{max} - F_i$, $i = 1 \dots 50$) на всех итерациях. После, по каждой задаче искалось медианное решение данной проблемы, и, наконец, все решения задач в датасете суммировались. Для более наглядного изменения значений по итерациям был взят \log_{10} с обратным знаком. Результаты представлены на графиках ниже для каждого датасета. По оси абсцисс отложен значение логарифма невязки ($-\log_{10} \sum_{i=1}^{50} (F_{max} - F_i)$)







Заключение и выводы

В данной работе был представлен новый эффективный алгоритм глобальной оптимизации, основанный на оценке константы Липшица неизвестной функции. Этот алгоритм является модификацией метода *AdaLIPO* с измененной стратегией принятия решения, а также сравнивался по эффективности с алгоритмом, упомянутым выше, идейным «предком» - *LIPO*, и методами, которые являются классическими в области решения проблемы глобальной оптимизации, – *CMA-ES* и Байесовская оптимизация.

Результаты показывают, что представленный алгоритм для тестовых данных всегда работает лучше, чем Байесовская оптимизация и *AdaLIPO*, и в 6 из 8 случаях лучше, чем *CMA-ES*. Данные показатели говорят о стабильной эффективности для рассматриваемого класса задач и меньшем пороге сходимости, чем у алгоритмов, поставленных ему в сравнение.

Тем не менее, анализ алгоритма не завершен с аналитической точки зрения, так как доказана лишь его сходимость.

Список литературы

- [1] Zhou Q. et al. Optimization of laser brazing onto galvanized steel based on ensemble of metamodels //Journal of Intelligent Manufacturing. – 2018. – Т. 29. – С. 1417-1431.
- [2] Jiang C. et al. An active failure-pursuing Kriging modeling method for time-dependent reliability analysis //Mechanical Systems and Signal Processing. – 2019. – Т. 129. – С. 112-129.
- [3] Dong H., Song B., Wang P. Kriging-based optimization design for a new style shell with black box constraints //Journal of Algorithms & Computational Technology. – 2017. – Т. 11. – №. 3. – С. 234-245.
- [4] White D. A. et al. Multiscale topology optimization using neural network surrogate models //Computer Methods in Applied Mechanics and Engineering. – 2019. – Т. 346. – С. 1118-1135.
- [5] Liu H., Ong Y. S., Cai J. A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design //Structural and Multidisciplinary Optimization. – 2018. – Т. 57. – С. 393-416.
- [6] Shojaeefard M. H., Hosseini S. E., Zare J. CFD simulation and Pareto-based multi-objective shape optimization of the centrifugal pump inducer applying GMDH neural network, modified NSGA-II, and TOPSIS //Structural and Multidisciplinary Optimization. – 2019. – Т. 60. – С. 1509-1525.
- [7] Dong H. et al. Multi-surrogate-based Differential Evolution with multi-start exploration (MDEME) for computationally expensive optimization //Advances in Engineering Software. – 2018. – Т. 123. – С. 62-76.
- [8] Stander J. N., Venter G., Kamper M. J. High fidelity multidisciplinary design optimisation of an electromagnetic device //Structural and Multidisciplinary Optimization. – 2016. – Т. 53. – С. 1113-1127.
- [9] Lawler E. L. A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem //Management science. – 1972. – Т. 18. – №. 7. – С. 401-405.
- [10] Sayadi M. K., Hafezalkotob A., Naini S. G. J. Firefly-inspired algorithm for discrete optimization problems: An application to manufacturing cell formation //Journal of Manufacturing Systems. – 2013. – Т. 32. – №. 1. – С. 78-84.
- [11] Ekel P. Y., Neto F. H. S. Algorithms of discrete optimization and their application to problems with fuzzy coefficients //Information Sciences. – 2006. – Т. 176. – №. 19. – С. 2846-2868.
- [12] Dede T. Application of teaching-learning-based-optimization algorithm for the discrete optimization of truss structures //Ksce journal of civil engineering. – 2014. – Т. 18. – С. 1759-1767.
- [13] Bartz-Beielstein T., Zaefferer M. Model-based methods for continuous and discrete global optimization //Applied Soft Computing. – 2017. – Т. 55. – С. 154-167.
- [14] Land A. H., Doig A. G. An automatic method for solving discrete programming problems. – Springer Berlin Heidelberg, 2010. – С. 105-132.
- [15] Demeulemeester E., Herroelen W. A branch-and-bound procedure for the multiple resource-constrained project scheduling problem //Management science. – 1992. – Т. 38. – №. 12. – С. 1803-1818.

- [16] Nakariyakul S., Casasent D. P. Adaptive branch and bound algorithm for selecting optimal features //Pattern Recognition Letters. – 2007. – T. 28. – №. 12. – C. 1415-1427.
- [17] Mladenović N., Hansen P. Variable neighborhood search //Computers & operations research. – 1997. – T. 24. – №. 11. – C. 1097-1100.
- [18] Adibi M. A., Zandieh M., Amiri M. Multi-objective scheduling of dynamic job shop using variable neighborhood search //Expert Systems with Applications. – 2010. – T. 37. – №. 1. – C. 282-287.
- [19] Kytöjoki J. et al. An efficient variable neighborhood search heuristic for very large scale vehicle routing problems //Computers & operations research. – 2007. – T. 34. – №. 9. – C. 2743-2757.
- [20] Polacek M. et al. A variable neighborhood search for the multi depot vehicle routing problem with time windows //Journal of heuristics. – 2004. – T. 10. – C. 613-627.
- [21] Abramson M. A. et al. Mesh adaptive direct search algorithms for mixed variable optimization //Optimization Letters. – 2009. – T. 3. – C. 35-47.
- [22] Anghinolfi D., Paolucci M. A new discrete particle swarm optimization approach for the single-machine total weighted tardiness scheduling problem with sequence-dependent setup times //European Journal of Operational Research. – 2009. – T. 193. – №. 1. – C. 73-85.
- [23] Zhang S. et al. Swarm intelligence applied in green logistics: A literature review //Engineering Applications of Artificial Intelligence. – 2015. – T. 37. – C. 154-169.
- [24] Guendouz M., Amine A., Hamou R. M. A discrete modified fireworks algorithm for community detection in complex networks //Applied Intelligence. – 2017. – T. 46. – C. 373-385.
- [25] Li X. et al. A novel complex network community detection approach using discrete particle swarm optimization with particle diversity and mutation //Applied Soft Computing. – 2019. – T. 81. – C. 105476.
- [26] Hasançebi O., Carbas S. Bat inspired algorithm for discrete size optimization of steel frames //Advances in Engineering Software. – 2014. – T. 67. – C. 173-185.
- [27] Gholizadeh S., Razavi N., Shojaei E. Improved black hole and multiverse algorithms for discrete sizing optimization of planar structures //Engineering Optimization. – 2019. – T. 51. – №. 10. – C. 1645-1667.
- [28] Younis A., Dong Z. Trends, features, and tests of common and recently introduced global optimization methods //Engineering Optimization. – 2010. – T. 42. – №. 8. – C. 691-718.
- [29] Dong H. et al. Multi-surrogate-based global optimization using a score-based infill criterion //Structural and Multidisciplinary Optimization. – 2019. – T. 59. – C. 485-506.
- [30] Dong H. et al. SCGOSR: Surrogate-based constrained global optimization using space reduction //Applied Soft Computing. – 2018. – T. 65. – C. 462-477.
- [31] Zhou Q. et al. A two-stage adaptive multi-fidelity surrogate model-assisted multi-objective genetic algorithm for computationally expensive problems //Engineering with computers. – 2021. – T. 37. – C. 623-639.

- [32] Shi R. et al. Multi-fidelity modeling and adaptive co-kriging-based optimization for all-electric geostationary orbit satellite systems //Journal of Mechanical Design. – 2020. – T. 142. – №. 2.
- [33] Sacks J. et al. Design and analysis of computer experiments //Statistical science. – 1989. – T. 4. – №. 4. – C. 409-423.
- [34] Hardy R. L. Multiquadric equations of topography and other irregular surfaces //Journal of geophysical research. – 1971. – T. 76. – №. 8. – C. 1905-1915.
- [35] Hosder S. et al. Polynomial response surface approximations for the multidisciplinary design optimization of a high speed civil transport //Optimization and Engineering. – 2001. – T. 2. – C. 431-452.
- [36] Gu J., Zhang H., Zhong X. Hybrid meta-model-based global optimum pursuing method for expensive problems //Structural and Multidisciplinary Optimization. – 2020. – T. 61. – №. 2. – C. 543-554.
- [37] Tsoulos I. G. Modifications of real code genetic algorithm for global optimization //Applied Mathematics and Computation. – 2008. – T. 203. – №. 2. – C. 598-607.
- [38] Fraley C., Raftery A. E. Model-based clustering, discriminant analysis, and density estimation //Journal of the American statistical Association. – 2002. – T. 97. – №. 458. – C. 611-631.
- [39] Martinez-Cantin R. Bayesopt: A bayesian optimization library for nonlinear optimization, experimental design and bandits //J. Mach. Learn. Res. – 2014. – T. 15. – №. 1. – C. 3735-3739.
- [40] Shubert B. O. A sequential method seeking the global maximum of a function //SIAM Journal on Numerical Analysis. – 1972. – T. 9. – №. 3. – C. 379-388.
- [41] Piyavskii S. A. An algorithm for finding the absolute extremum of a function //USSR Computational Mathematics and Mathematical Physics. – 1972. – T. 12. – №. 4. – C. 57-67.
- [42] Jones D. R., Perttunen C. D., Stuckman B. E. Lipschitzian optimization without the Lipschitz constant //Journal of optimization Theory and Applications. – 1993. – T. 79. – C. 157-181.
- [43] Huyer W., Neumaier A. Global optimization by multilevel coordinate search //Journal of Global Optimization. – 1999. – T. 14. – C. 331-355.
- [44] Valko M., Carpentier A., Munos R. Stochastic simultaneous optimistic optimization //International Conference on Machine Learning. – PMLR, 2013. – C. 19-27.
- [45] Preux P., Munos R., Valko M. Bandits attack function optimization //2014 IEEE congress on evolutionary computation (CEC). – IEEE, 2014. – C. 2245-2252.
- [46] Munos R. et al. From bandits to monte-carlo tree search: The optimistic principle applied to optimization and planning //Foundations and Trends in Machine Learning. – 2014. – T. 7. – №. 1. – C. 1-129.
- [47] Malherbe C., Contal E., Vayatis N. A ranking approach to global optimization //International Conference on Machine Learning. – PMLR, 2016. – C. 1539-1547.
- [48] Grill J. B., Valko M., Munos R. Black-box optimization of noisy functions with unknown smoothness //Advances in Neural Information Processing Systems. – 2015. – T. 28.

- [49] Bull A. D. Convergence rates of efficient global optimization algorithms //Journal of Machine Learning Research. – 2011. – T. 12. – №. 10.
- [50] Malherbe C., Vayatis N. Global optimization of Lipschitz functions //International Conference on Machine Learning. – PMLR, 2017. – C. 2314-2323.
- [51] Dasgupta S. Two faces of active learning //Theoretical computer science. – 2011. – T. 412. – №. 19. – C. 1767-1781.
- [52] Hanneke S. Rates of convergence in active learning //The Annals of Statistics. – 2011. – C. 333-361.
- [53] Malherbe C., Contal E., Vayatis N. A ranking approach to global optimization //International Conference on Machine Learning. – PMLR, 2016. – C. 1539-1547.
- [54] Wood G. R., Zhang B. P. Estimation of the Lipschitz constant of a function //Journal of Global Optimization. – 1996. – T. 8. – C. 91-103.
- [55] Bubeck S., Stoltz G., Yu J. Y. Lipschitz bandits without the lipschitz constant //Algorithmic Learning Theory: 22nd International Conference, ALT 2011, Espoo, Finland, October 5-7, 2011. Proceedings 22. – Springer Berlin Heidelberg, 2011. – C. 144-158.
- [56] Hansen N. The CMA evolution strategy: a comparing review //Towards a new evolutionary computation: Advances in the estimation of distribution algorithms. – 2006. – C. 75-102.