

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Нижегородский государственный университет
им. Н.И. Лобачевского»

Институт информационных технологий, математики и механики

**Кафедра математического обеспечения и суперкомпьютерных
технологий**

ОТЧЁТ

по научно-исследовательской практике

Название работы

Выполнил: студент группы М0813-2

_____ Соврасов В.В.

Подпись

Научный руководитель:

доцент, к.ф.м.н.

_____ Баркалов К.А.

Подпись

Нижний Новгород
2016

Содержание

1. Введение	2
2. Алгоритм глобального поиска	2
2.1. Сравнение методов оптимизации	3
3. Многоуровневая схема редукции размерности с помощью разверток	4
4. Сравнение различных типов множественных разверток	6
5. Применение локального поиска для ускорения сходимости АГП	7
6. Смешанный алгоритм глобального поиска и его эффективная реализация	7
Список литературы	8

Введение

//здесь про важность задач глобальной оптимизации Задачи глобальной оптимизации встречаются в . Сложность этих задач экспоненциально растёт в зависимости от размерности пространства поиска, поэтому для решения существенно многомерных задач требуются суперкомпьютерные вычисления.

В настоящее время на кафедре МОиСТ активно ведётся разработка программной системы для глобальной оптимизации функций многих вещественных переменных ExaMin. Эта система включает в себя последние теоретические разработки, сделанные на кафедре в этой сфере, в том числе и блочную многошаговую схему редукции размерности [1]. Отличительной чертой системы является то, что, она может работать как на CPU, так на разных типах ускорителей вычислений с высокой степенью параллельности (XeonPhi, GPU Nvidia) [2].

В данной работе будут описаны некоторые улучшения, внесённые в систему, и предварительные исследования, проведённые перед их внедрением.

Алгоритм глобального поиска

Поэтому, не ограничивая общности, можно рассматривать минимизацию одномерной функции $f(x) = \phi(y(x))$, $x \in [0, 1]$, удовлетворяющей условию Гельдера.

Рассматриваемый алгоритм решения данной задачи предполагает построение последовательности точек x_k , в которых вычисляются значения минимизируемой функции $z_k = f(x_k)$. Процесс вычисления значения функции (включающий в себя построение образа $y_k = y(x_k)$) будем называть испытанием, а пару (x_k, z_k) — результатом испытания. Множество пар $\{(x_k, z_k)\}$, $1 \leq k \leq n$ составляют поисковую информацию, накопленную методом после проведения n шагов. В нашем распоряжении имеется $p \geq 1$ вычислительных элементов и в рамках одной итерации метода мы будем проводить p испытаний одновременно. Обозначим $k(n)$ общее число испытаний, выполненных после n параллельных итераций.

На первой итерации метода испытание проводится в произвольной внутренней точке x_1 интервала $[0, 1]$. Пусть выполнено $n \geq 1$ итераций метода, в процессе которых были проведены испытания в $k = k(n)$ точках x_i , $1 \leq i \leq k$. Тогда точки x^{k+1}, \dots, x^{k+p} поисковых испытаний следующей $(n + 1)$ -ой итерации определяются в соответствии с правилами:

Шаг 1. Перенумеровать точки множества $X_k = \{x^1, \dots, x^k\} \cup \{0\} \cup \{1\}$, которое включает в себя граничные точки интервала $[0, 1]$, а также точки предшествующих испытаний, нижними индексами в порядке увеличения значений координаты, т.е.

$$0 = x_0 < x_1 < \dots < x_{k+1} = 1$$

Шаг 2. Полагая $z_i = f(x_i)$, $1 \leq i \leq k$, вычислить величины

$$\mu = \max_{1 \leq i \leq k} \frac{|z_i - z_{i-1}|}{\Delta_i}, M = \begin{cases} r\mu, \mu > 0 \\ 1, \mu = 0 \end{cases} \quad (2.1)$$

где r является заданным параметром метода, а $\Delta_i = (x_i - x_{i-1})^{\frac{1}{N}}$.

Шаг 3. Для каждого интервала (x_{i-1}, x_i) , $1 \leq i \leq k+1$, вычислить характеристику в соответствии с формулами

$$R(1) = 2\Delta_1 - 4\frac{z_1}{M}, R(k+1) = 2\Delta_{k+1} - 4\frac{z_k}{M} \quad (2.2)$$

$$R(i) = \Delta_i + \frac{(z_i - z_{i-1})^2}{M^2\Delta_i} - 2\frac{z_i + z_{i-1}}{M}, 1 < i < k+1 \quad (2.3)$$

Шаг 4. Характеристики $R(i)$, $1 \leq i \leq k+1$, упорядочить в порядке убывания

$$R(t_1) \geq R(t_2) \geq \dots \geq R(t_k) \geq R(t_{k+1}) \quad (2.4)$$

и выбрать p наибольших характеристик с номерами интервалов t_j , $1 \leq j \leq p$.

Шаг 5. Провести новые испытания в точках x_{k+j} , $1 \leq j \leq p$, вычисленных по формулам

$$x_{k+j} = \frac{x_{t_j} + x_{t_j-1}}{2}, t_j = 1, t_j = k+1$$

$$x_{k+j} = \frac{x_{t_j} + x_{t_j-1}}{2} - (z_{t_j} - z_{t_j-1}) \frac{1}{2r} \left[\frac{|z_{t_j} - z_{t_j-1}|}{\mu} \right]^N, 1 < t_j < k+1 \quad (2.5)$$

Алгоритм прекращает работу, если выполняется условие $\Delta_{t_j} \leq \varepsilon$ хотя бы для одного номера t_j , $1 \leq j \leq p$; здесь $\varepsilon > 0$ есть заданная точность. В качестве оценки глобально-оптимального решения задачи выбираются значения

$$f_k^* = \min_{1 \leq i \leq k} f(x_i), x_k^* =_{1 \leq i \leq k} f(x_i) \quad (2.6)$$

Теоретическое обоснование данного способа организации параллельных вычислений изложено в [strGergrParOptBook].

Сравнение методов оптимизации

Существует несколько критериев оптимальности алгоритмов поиска (минимаксный, критерий одношаговой оптимальности), но большинстве случаев представляет интерес сравнение методов по среднему результату, достижимому на конкретном подклассе липшицевых функций. Достоинством такого подхода является то, что средний показатель можно оценить по конечной случайной выборке задач, используя методы математической статистики.

В качестве оценки эффективности алгоритма будем использовать, операционную характеристику, которая определяется множеством точек на плоскости (K, P) , где K – среднее

число поисковых испытаний, предшествующих выполнению условия останова при минимизации функции из данного класса, а P – статистическая вероятность того, что к моменту останова глобальный экстремум будет найден с заданной точностью. Если при выбранном K операционная характеристика одного метода лежит выше характеристики другого, то это значит, что при фиксированных затратах на поиск первый метод найдёт решение с большей статистической вероятностью. Если же зафиксировать некоторое значение P , и характеристика одного метода лежит левее характеристики другого, то первый метод требует меньше затрат на достижение той же надёжности.

Многоуровневая схема редукции размерности с помощью разверток

Одна из постановок задачи глобальной оптимизации звучит следующим образом: найти глобальный минимум N -мерной функции $\phi(y)$ в гиперинтервале $D = \{y \in R^N : a_i \leq x_i \leq b_i, 1 \leq i \leq N\}$. Для построения оценки глобального минимума по конечному количеству вычислений значения функции требуется, чтобы $\phi(y)$ удовлетворяла условию Липшица.

$$\phi(y^*) = \min\{\phi(y) : y \in D\} \quad (3.7)$$

$$|\phi(y_1) - \phi(y_2)| \leq L\|y_1 - y_2\|, y_1, y_2 \in D, 0 < L < \infty \quad (3.8)$$

Классической схемой редукции размерности для алгоритмов глобальной оптимизации является использование разверток — кривых, заполняющих пространство [3].

$$\{y \in R^N : -2^{-1} \leq y_i \leq 2^{-1}, 1 \leq i \leq N\} = \{y(x) : 0 \leq x \leq 1\}$$

Такое отображение позволяет свести задачу в многомерном пространстве к решению одномерной ценой ухудшения её свойств. В частности, одномерная функция $\phi(y(x))$ является не Липшицевой, а Гёльдеровой:

$$|\phi(y(x_1)) - \phi(y(x_2))| \leq H|x_1 - x_2|^{\frac{1}{N}}, x_1, x_2 \in [0, 1]$$

где константа Гельдера H связана с константой Липшица L соотношением

$$H = 4Ld\sqrt{N}, d = \max\{b_i - a_i : 1 \leq i \leq N\}$$

Теоретически с помощью этой схемы можно решить задачу любой размерности, однако на ЭВМ развертка строится с помощью конечноразрядной арифметики, из-за чего начиная с некоторого N^* построение разветки невозможно (значение N^* зависит от максимального количества значащих разрядов в арифметике с плавающей точкой). Понять почему это происходит нетрудно, обратившись, например к [3].

Чтобы преодолеть эту проблему профессором В. П. Гергелем была предложена следующая идея: использовать композицию разверток меньшей размерности для построения отображения $z(x) : [0; 1] \rightarrow D \in R^N$. Поясним эту схему на примере редукции размерности в

четырёхмерной задаче. Пусть $y_2(x)$ — двумерная развертка (отображает отрезок в прямоугольник), тогда рассмотрим функцию $\psi(x_1, x_2) = \phi(y_2(x_1), y_2(x_2))$. К $\psi(x_1, x_2)$ можно также применить редукцию размерности с помощью развертки. Таким образом, задав точку $x^* \in [0; 1]$, вычислив $y_2(x^*) = (x_1, x_2)$ и пару векторов $(y_2(x_1), y_2(x_2))$, получим четырёхмерную точку. Из инъективности $y_2(x)$ следует инъективность $z(x)$.

Проблемой этого метода является выяснение свойств функции $\phi(z(x))$ и возможности использования одномерного метода Стронгина с гёльдеровой метрикой для её оптимизации. Чтобы не тратить время на теоретическое исследование, были проведены численные эксперименты с целью оценить возможности применения многоуровневой развёртки в четырёхмерном случае.

Прежде всего, рассмотрим линии уровня функции $\psi(x_1, x_2) = \phi(y_2(x_1), y_2(x_2))$ при $\phi(t) = \sum_{i=1}^4 (t_i - 0.5)^2$. Как видно из рис. 3.1, линии уровня имеют довольно сложную струк-

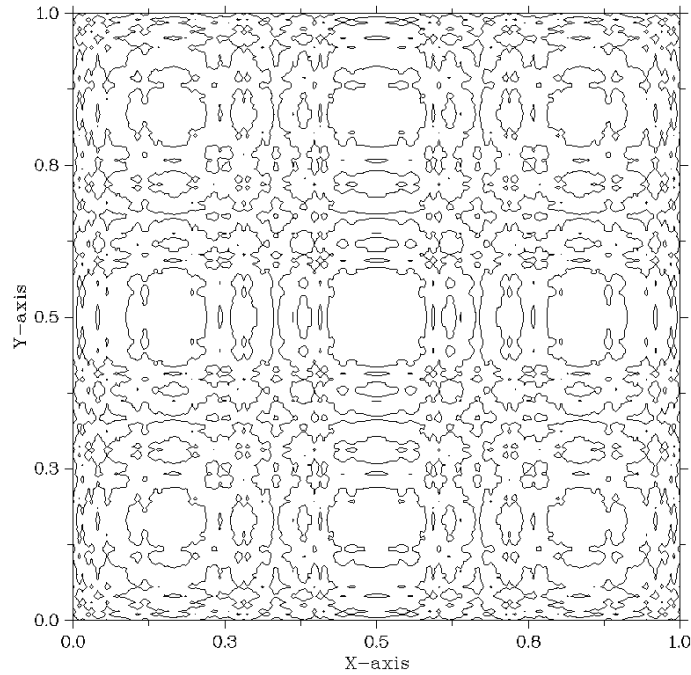


Рис. 3.1: Линии уровня функции $\psi(x_1, x_2) = \phi(y_2(x_1), y_2(x_2))$

туру, что говорит о возможных сложностях применения одномерного метода с разверткой.

Далее был проведён более масштабный вычислительный эксперимент: с помощью многоуровневой развертки решались 100 задач из класса GKLS Simple 4d. На рис. 3.2 приведены операционные характеристики метода с простой и многоуровневой развёртками. При этом были зафиксированы следующие параметры алгоритма: надёжность $r = 4.5$, плотность построения всех развёрток 12, критерий остановки попадание точки, поставленной методом в квадрат со стороной $\varepsilon = 10^{-2}$, центром которого является решение задачи. Как видно из графика, операционная характеристика метода с многоуровневой развёрткой лежит выше, чем аналогичная кривая метода с простой развёрткой. Кроме того, метод с многоуровневой

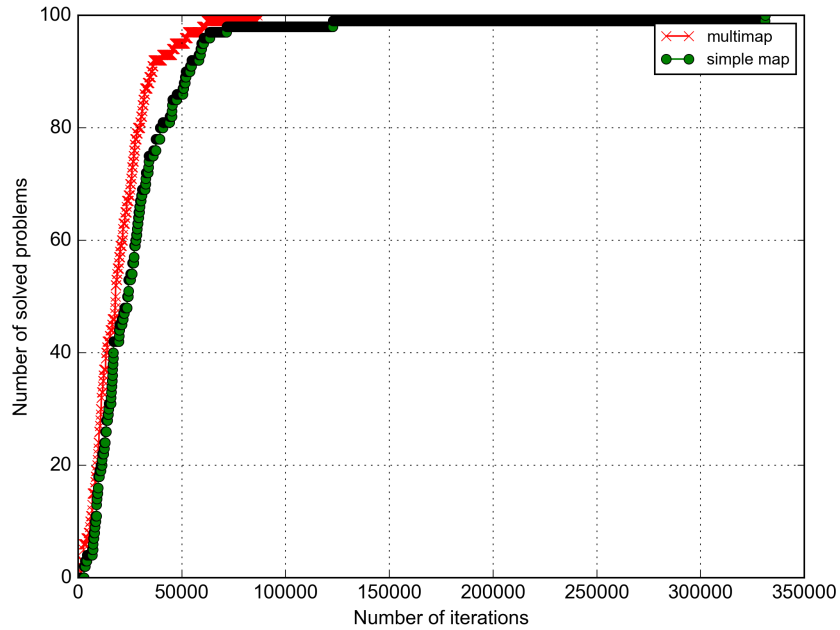


Рис. 3.2: Операционная характеристика метода с различными развёртками на классе GKLS Simple 4d

развёрткой заметно раньше вышел на стопроцентную надёжность на решаемой выборке задач. Исходя из первых экспериментов можно сказать, что при данных параметрах алгоритма многоуровневая развёртка лучше, но при уменьшении точности алгоритма ε до значения 10^{-3} выполнения критерия остановки в случае использования многоуровневой развёртки с выбранной плотностью построения внутренних развёрток не происходит. Если плотность увеличить до 16, то остановка также не будет происходить. Исходя из этого можно делать вывод, что использование на практике многоуровневых развёрток с большой долей вероятности невозможно из-за описанного дефекта сходимости.

Сравнение различных типов множественных разверток

Одним из недостатков использования метода редукции размерности с помощью развёрток является потеря информации о близости точек в многомерном пространстве при решении одномерной задачи. На рис. 4.3 представлена развёртка двухмерного параболоида вращения. Как видно из графика, единственный глобальный уровень расщепился на три. Это происходит из-за того, что развёртка может возвращаться в окрестность некоторых точек при разных (далёких друг от друга) значениях одномерного параметра. Чтобы частично компенсировать потерю информации о многомерной окрестности точки, были предложены множественные развёртки [1].

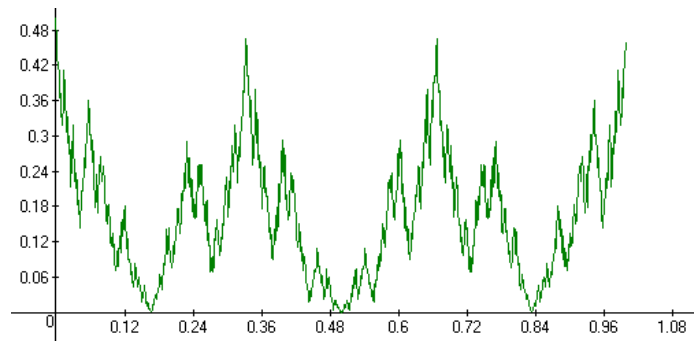


Рис. 4.3: Одномерная функция после подстановки развертки в $\phi(y) = y_1^2 + y_2^2$

Применение локального поиска для ускорения сходимости АГП

Методы локального поиска могут применяться в сочетании с глобальными алгоритмами для улучшения полученных решений или текущих оценок оптимума. В первом случае локальный метод стартует из точки, найденной глобальным методом, и уточняет решение практически до любой нужной точности. Это позволяет избежать чрезмерных затрат на поиск решения с высокой точностью глобальными методами.

Во втором случае локальный метод используется для ускорения обнаружения локальных оптимумов. Информационно-статистический метод Стронгина позволяет обновлять свою поисковую информацию из любых посторонних источников, в том числе и точки испытаний, полученные от локального метода. Как только глобальный метод находит новую оценку оптимума, из этой точки стартует локальный метод и все или часть испытаний, проведённых им добавляется в поисковую информацию, далее глобальный метод продолжает работу. Каких-либо теоретических исследований подобной схемы не проводилось, поэтому её эффективность проверялась эспериментально.

В качестве метода локальной оптимизации был выбран метод Хука-Дживса [4]. Он прост в реализации и относится к прямым методам — для его работы не требуется знать значений каких-либо производных оптимизируемой функции.

Были проведены две серии экспериментов, соответствующих следующим схемам добавления точек, полученных локальным методом в поисковую информацию:

- Добавление единственной точки, к которой сошёлся локальный метод.
- Добавление всех точек, поставленных локальным методом.

Эксперименты проводились на классе GKLS 4d Simple, параметры метода были заданы такие же, как в разделе 3.

Смешанный алгоритм глобального поиска и его эффективная реализация

Список литературы

- [1] В.П. Гергель А.В. Сысоев К.А. Баркалов. «Блочная многошаговая схема параллельного решения задач многомерной глобальной оптимизации». В: *Материалы XIV Международной конференции "Высокопроизводительные параллельные вычисления на кластерных системах 10-12 ноября, ПНИПУ, Пермь. 2014*, 425–432.
- [2] Сысоев А.В. Баркалов К.А. Гергель В.П. Лебедев И.Г.. «MPI-реализация блочной многошаговой схемы параллельного решения задач глобальной оптимизации». В: *Суперкомпьютерные дни в России: Труды международной конференции (28-29 сентября 2015 г., г. Москва)*. М.: Изд-во МГУ, 2015, с. 411—419.
- [3] Sergeyev Ya.D. Strongin R.G. *Global Optimization with Non-convex Constraints. Sequential and Parallel Algorithms*. Kluwer Academic Publishers, 2000, 704p.
- [4] Химмельблау Д. М.. *Прикладное нелинейное программирование*. Издательство МИР, Москва, 1975.