Использование параллельной системы глобальной оптимизации ExaMin для решения задач конкурса GenOpt[[1]](#footnote-2)\*

В.В. Соврасов

Нижегородский государственный университет им. Н.И. Лобачевского

В работе рассматриваются параллельные алгоритмы решения задач многоэкстремальной оптимизации. Алгоритмы разработаны в рамках развиваемого в ННГУ им. Н.И. Лобачевского информационно-статистического подхода и реализованы в параллельной системе ExaMin. Обсуждаются результаты вычислительных экспериментов, которые проводились на тестовых функциях международного конкурса алгоритмов оптимизации GENOPT (www.genopt.org).

*Ключевые слова:* глобальная оптимизация, многоэкстремальные функции, редукция размерности, параллельные алгоритмы.

# 1. Введение

Один из известных подходов к исследованию и сравнению алгоритмов многоэкстремальной оптимизации основан на применении этих методов для решения множества тестовых задач, выбираемых случайным образом из некоторого сконструированного класса. При этом каждая тестовая задача может рассматриваться как конкретная реализация случайной функции задаваемой с помощью специального генератора.

К числу таких генераторов для одномерных задач относятся выборки из рядов Фурье предложенные Хиллом [], генератор предложенный Шекелем []. Для исследования различных одномерных алгоритмов на случайных выборках функций порождаемых генераторами Хилла и Шекеля создана специальная программная система АБСОЛЮТ (GLOBALIZER) [-]..

Генератор для случайного выбора двухмерных тестовых функций, успешно примененный в исследованиях ряда авторов, был предложен Гришагиным []. Генератор для функций произвольной размерности с задаваемым расположением локальных и глобального минимумов предложен в работе [7]. Его использование для изучения некоторых многомерных алгоритмов описано в работах [8].

Указанный подход к сравнению алгоритмов был использован организаторами конкурса GENeralization-based challenge in global OPTimization (GENOPT). Целевые функции разделены на три семейства, GKLS, классические задачи и составные. Каждое семейство разделено на шесть классов со своим типом и размерностью. Каждая функция из класса получается случайным образом по некоторым параметрам.

В ННГУ им. Н.И. Лобачевского под руководством проф. Р.Г. Стронгина разработан эффективный подход к решению задач глобальной оптимизации [-]. В рамках данного подхода решение многомерных задач сводится к решению эквивалентных им одномерных. Соответствующая редукция основана на использовании кривых Пеано (называемых также развертками), однозначно отображающих единичный отрезок вещественной оси на гиперкуб, а также их обобщений, которые можно применять при решении задачи на многопроцессорных системах. Предложенные алгоритмы были реализованы в решателе ExaMin, примененном к решению задач конкурса GENOPT. В настоящей работе дано краткое описание использованного алгоритма глобальной оптимизации и его модификаций, а также приведены результаты численных экспериментов с конкурсными задачами.

# 2. Постановка задачи

Задача многомерной многоэкстремальной оптимизации может быть определена как проблема поиска наименьшего значения действительной функции *ϕ*(*y*)



где *a*,*b*∈*RN* есть заданные векторы.

Численное решение задачи (1) сводится к построению оценки , отвечающей некоторому понятию близости к точке  на основе конечного числа k вычислений значений оптимизируемой функции. Относительно класса рассматриваемых задач предполагается, что *ϕ*(*y*) удовлетворяет условию Липшица



что соответствует ограниченности изменения значений функции при ограниченной вариации аргумента.

Таблица . Классы конкурсных задач

|  |  |  |  |
| --- | --- | --- | --- |
| Семейство | Класс | Размерность | Название |
| GKLS | не дифференцируемые | 10 | GKLS\_nd\_10 |
| 30 | GKLS\_nd\_30 |
| дифференцируемые | 10 | GKLS\_cd\_10 |
| 30 | GKLS\_cd\_30 |
| дважды дифференцируемые | 10 | GKLS\_td\_10 |
| 30 | GKLS\_td\_30 |
| Классические | Розенброк | 10 | Rosenbrock\_10 |
| 30 | Rosenbrock\_30 |
| Растригин | 10 | Rastrigin\_10 |
| 30 | Rastrigin\_30 |
| Захаров | 10 | Zakharov\_10 |
| 30 | Zakharov\_30 |
| Составные(по три класса каждой размерности) | | 10 | Composite\_10 |
| 30 | Composite\_30 |

Организаторами конкурса GENOPT были предложены 18 классов задач (см. таблицу 1). В каждом классе было задано 100 функций. Первое семейство состоит из функций полученных GKLS-генератором [, ], позволяющий порождать задачи многоэкстремальной оптимизации с заранее известными свойствами: количеством локальных минимумов, размерами их областей притяжения, точкой глобального минимума, значением функции в ней и т.п. Было выбрано три класса задач: не дифференцируемые, дифференцируемые и дважды дифференцируемые функции. Для всех функций было задано: пять локальных минимумов, радиус притяжения глобального минимума 1/3, distance of global minimizer r = 2=3; Задачи внутри класса отличаются расположением и величиной минимумов.

Второе семейство основано на классических непрерывных функциях. К ним применено преобразование которое растягивает функцию по отдельным направлениям. Использованы функции: Розенброка, Растригина и Захарова.

Функции из третьего семейства получены с помощью суммы случайного количества классических(базовых) непрерывных функций: Goldstein-Price – размерность 2, Hartmann - размерность 3 или 6, Розенброк, Растригин, Sphere, Захоров – размерность определяется случайно в диапазоне от 3 до половины размености получаемой функции. Каждая базовая функция принимает часть вектора координат(без пересечений), при этом сумма размерностей всех базовых функций равна размерности получаемой функции.

# 3. Используемые методы

В данном разделе описан базовый параллельный алгоритм глобального поиска(АГП) и его модификации, позволяющие решать многомерные задачи, использованные при решении различных конкурсных задач. Главной модификацией является использование редукция размерности с использованием кривых Пеано, позволяющая решать алгоритмом глобального поиска многомерные задачи. Следующей модификацией является использование локального метода. Обе модификации применяются для всех задач.

Еще одним способом редукции размерности является блочная многошаговая схема, эта модификация позволяет использовать множество процессов для решения многомерных задач. В конкурсе не использовалась, поскольку является параллельным алгоритмом.

Многие задачи из 10-мерных классов GKLS решаются с использованием первых двух модификаций, однако для 30-мерных задач этого оказалось недостаточно. Следующим вариантом модификации стала «мультистартовая» схема. Используя генератор псевдослучайных чисел Соболя, мы выбирали 900 начальных точек, в каждой из которых запускался локальный метод оптимизации не только с ограничением по точности и по числу итераций (не более 1000). Таким образом, в худшем случае на локальную фазу могло быть потрачено 900 тыс. испытаний. Все точки выполненных на локальной фазе испытаний помещались в базу глобального метода. Далее запускалась глобальная фаза, на которой оставшиеся до 1 миллиона испытания выполняет АГП.

Поскольку функция Растригина сепарабельна, ее глобальный оптимум можно искать, выполняя оптимизацию по каждой координате в отдельности. Таким образом, для решения задачи Растригина использовалась следующая схема:

* Сепарабельная фаза
  + Выбрать начальную точку.
  + Для каждой координаты выполнить:
    - Зафиксировать все координаты, кроме текущей.
    - Выполнить оптимизацию одномерным АГП.
  + Все точки испытаний локальной метода добавлять в базу точек АГП.
* Локальная фаза
  + Запустить из точки текущего рекорда локального метод Хука-Дживса с заданным условием выхода по точности.
  + Все точки испытаний локальной метода добавлять в базу точек АГП.
  + По достижении локальным методом заданной точности перейти к Глобальной фазе.
* Глобальная фаза
  + Выполнять итерации АГП, пока не будет обновлено текущее значение глобального минимума.
  + При обновлении глобального минимума перейти к Локальной фазе.

Рассмотренная схема позволила решить все задачи с функцией Растригина как 10-ти, так и 30-мерные.

Этот же подход был использован и для решения задач, построенных на унимодальных функциях (Розенброка и Захарова). В этих случая сепарабельная фаза дает хорошее начальное приближение для локального метода. Без использования сепарабельности локальный метод стартует из далекой от минимума точки и делает слишком много итераций до достижения условия остановки.

## 3.1 Алгоритм глобального поиска

В качестве базовой задачи мы будет рассматривать одномерную задачу многоэкстремальной оптимизации , в которой целевая функция ϕ(x) удовлетворяет условию Липшица. Дадим детальное описание параллельного алгоритма глобального поиска (ПАГП), применяемого к ее решению.

Пусть в нашем распоряжении имеется  вычислительных элементов. Тогда на данной итерации можно провести одновременно p испытаний. Тогда общее число испытаний, выполненных после n параллельных итераций, составит .

Предположим, что выполнено  итераций метода (в качестве точек  первой итерации выбираются произвольные различные точки отрезка [0,1]). Тогда точки  текущей (*n+1*)-ой итерации определяются по следующим правилам.

Правило 1. Перенумеровать точки множества  так что 

Правило 2. Полагая , вычислить величины

, ,

где  является заданным параметром метода (параметр надежности), а .

Правило 3. Для каждого интервала , вычислить характеристику в соответствии с формулами

 ;

, 

Правило 4. Характеристики , упорядочить в порядке убывания



и выбрать p наибольших характеристик с номерами интервалов .

Правило 5. Провести новые испытания в точках , вычисленных по формулам

, ,  , .

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

, 

Данный способ организации параллельных вычислений имеет следующее обоснование [, ]. Используемые в алгоритме характеристики интервалов могут рассматриваться как некоторые меры вероятности локализации в данных интервалах точки глобального минимума. Неравенства упорядочивают интервалы по их характеристикам, и испытания проводятся параллельно в первых p интервалах, имеющих наибольшие вероятности. Различные модификации данного алгоритма и соответствующая теория сходимости представлены в [28].

## 3.2 Редукция размерности с использованием кривых Пеано

Для снижения сложности алгоритмов глобальной оптимизации, формирующих неравномерное покрытие области поиска, широко используются различные схемы редукции размерности, которые позволяют свести решение многомерных оптимизационных задач к семейству задач одномерной оптимизации.

Первым из рассматриваемых способов редукции размерности является использование кривой Пеано y(x), однозначно отображающей отрезок вещественной оси [0,1] на n-мерный куб

,

Вопросы численного построения отображений типа кривой Пеано и соответствующая теория подробно рассмотрены в []. Здесь же отметим, что численно построенная развертка является приближением к теоретической кривой Пеано с точностью порядка , где m – параметр построения развертки.

Использование подобного рода отображений позволяет свести многомерную задачу к одномерной задаче 

Важным свойством является сохранение ограниченности относительных разностей функции: если функция ϕ(y) в области D удовлетворяла условию Липшица с константой L, то функция ϕ(y(x)) на интервале [0,1] будет удовлетворять равномерному условию Гельдера

, ,

где константа Гельдера H связана с константой Липшица L соотношением

, .

Соотношение позволяет модифицировать приведенный в разделе 2 алгоритм решения одномерных задач для решения многомерных задач, редуцированных к одномерным. Для этого длины интервалов , участвующие в правилах (3)− алгоритма, заменяются на длины в новой метрике

,

а вместо формулы вводится выражение

, .

## 3.3 Рекурсивная схема редукции размерности

Схема рекурсивной оптимизации основана на известном (см. [29]) соотношении

,

которое позволяет заменить решение многомерной задачи решением семейства одномерных подзадач, рекурсивно связанных между собой.

Введем в рассмотрение множество функций

,

, .

Тогда, в соответствии с соотношением , решение исходной задачи сводится к решению одномерной задачи

.

Однако при этом каждое вычисление значения одномерной функции  в некоторой фиксированной точке предполагает решение одномерной задачи минимизации

,

и так далее до вычисления  согласно .

Для изложенной выше рекурсивной схемы предложено обобщение (блочная рекурсивная схема), которое комбинирует использование разверток и рекурсивной схемы с целью эффективного распараллеливания вычислений.

Рассмотрим вектор *y* как вектор блочных переменных

,

где *i*-я блочная переменная *ui* представляет собой вектор размерности  из последовательно взятых компонент вектора y, т.е. , ,…, , причем .

С использованием новых переменных основное соотношение многошаговой схемы может быть переписано в виде

,

где подобласти , являются проекциями исходной области поиска D на подпространства, соответствующие переменным .

Формулы, определяющие способ решения задачи на основе соотношений в целом совпадают с рекурсивной схемой −. Требуется лишь заменить исходные переменные , на блочные переменные .

При этом принципиальным отличием от исходной схемы является тот факт, что в блочной схеме вложенные подзадачи

, ,

являются многомерными, и для их решения может быть применен способ редукции размерности на основе кривых Пеано.

Число векторов и количество компонент в каждом векторе являются параметрами блочной многошаговой схемы и могут быть использованы для формирования подзадач с нужными свойствами. Например, если , т.е. , то блочная схема идентична исходной; каждая из вложенных подзадач является одномерной. А если , т.е. , то решение задачи эквивалентно ее решению с использованием единственной развертки, отображающей [0,1] в D; вложенные подзадачи отсутствуют.

## 3.4 Метод Хука Дживса

Еще одна реализованная модификация базового метода состояла в прямом использовании метода локальной оптимизации, а именно метода Хука-Дживса [] (см, также [, ]). Схематически работа комбинированного метода такова:

* Глобальная фаза
  + Выполнять итерации GSA, пока не будет обновлен текущий «рекорд» (минимальное значение целевой функции в уже посчитанных точках испытаний).
* Локальная фаза
  + Запустить из точки текущего рекорда локального метод Хука-Дживса с заданным условием выхода по точности.
  + Все точки испытаний локальной метода добавлять в базу точек GSA.
  + По достижении локальным методом заданной точности перейти к Глобальной фазе.

Точность локального метода (условие остановки) для всех классов задач GKLS, а также Растригина и композитных была взята равной 0.00001, для функции Розенброка – 0.0000005, для функции Захарова – 0.000001.

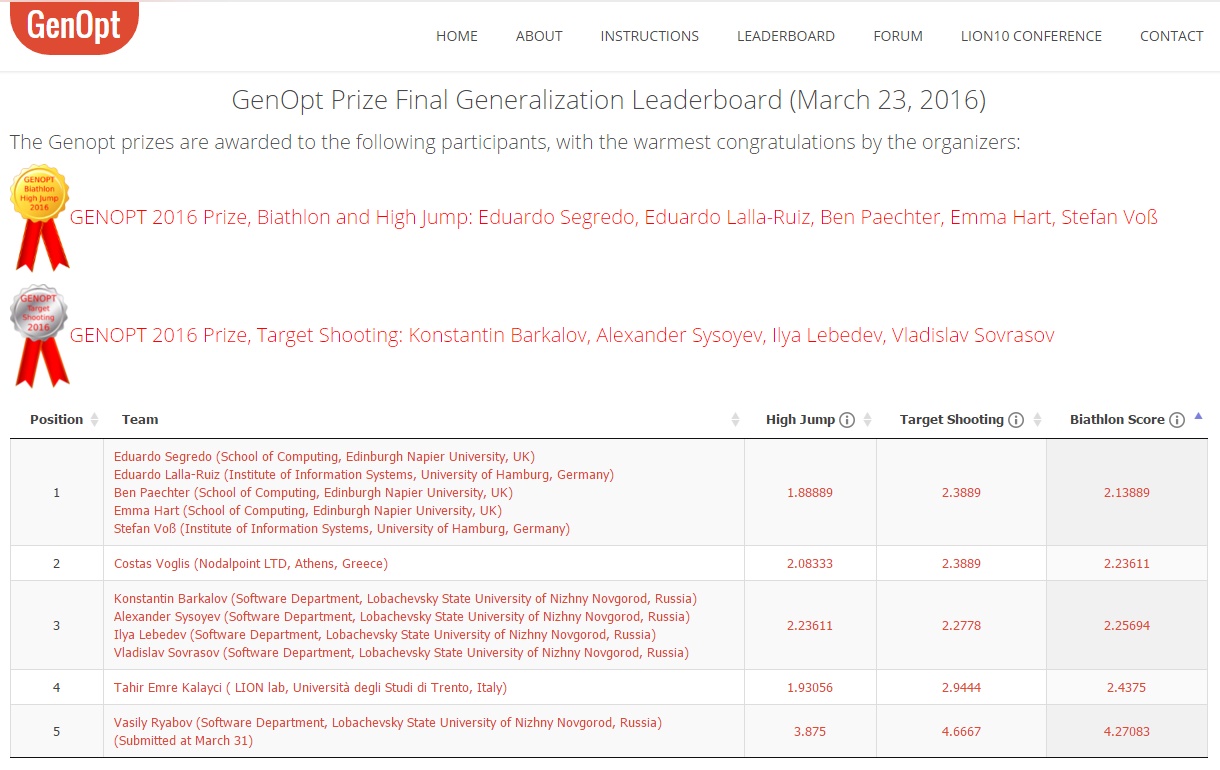
# 4. Результаты вычислительных экспериментов

Вычислительные эксперименты проводились на одном из узлов высокопроизводительного кластера ННГУ им. Н.И. Лобачевского. Узел кластера располагает 2-я процессорами Intel Xeon L5630 2.13 GHz, 24 Gb RAM. Центральный процессор является 4-х ядерным

Рассмотренные в разделе 3 методы и их модификации реализованы в решателе ExaMin, предназначенном для параллельного решения многомерных многоэкстремальных задач глобальной оптимизации, разрабатываемом в ННГУ им. Н.И. Лобачевского. Алгоритмическую основу решателя ExaMin составляют алгоритм глобального поиска и блочная многошаговая схема редукции размерности. По условиям конкурса при решении задач использовался только последовательный режим работы решателя. Однако ExaMin поддерживает работу как на системах с распределенной памятью (используя MPI), так и с общей памятью (использую OpenMP). Кроме того, поддерживаются графические процессоры NVIDIA и сопроцессоры Intel Xeon Phi.

В финальной части конкурса решатель ExaMin занял 3-е место в общем зачете и 1-е по общему числу решенных задач (рис. 1).

Распределение решенных задач по классам представлено в таблице 2. Во всех запусках точность построения развертки m равнялась 10. Параметр метода r был равен 2.5 для всех задач, кроме классов GKLS. Для GKLS r варьировался от 2.5 до 200.



**Рис. 1**. Final GenOpt 2016 leaderboard

Таблица 2. Процент решенных задач по классам

|  |  |
| --- | --- |
| Класс | Решилось |
| GKLS\_nd\_10 | 99 |
| GKLS\_nd\_30 | 15 |
| GKLS\_cd\_10 | 96 |
| GKLS\_cd\_30 | 1 |
| GKLS\_td\_10 | 94 |
| GKLS\_td\_30 | 0 |
| Rosenbrock\_10 | 100 |
| Rosenbrock\_30 | 100 |
| Rastrigin\_10 | 100 |
| Rastrigin\_30 | 100 |
| Zakharov\_10 | 100 |
| Zakharov\_30 | 100 |
| Composite\_10 | 100 |
| Composite\_30 | 100 |

Для всех классов, кроме GKLS использовалась модификация с сепарабельной фазой, с точностью 0.02.

При решении задач GKLS сначала использовались базовая версия алгоритма глобального поиска с редукцией размерности с использованием кривой Пеано и локальное уточнение. Затем для нерешенных задач вначале запускается случайный поиск. Распределение числа решенных задач без использования и с использованием случайного поиска указано в таблице 3.

Таблица 3. Использование случайного поиска при решении задач классов GKLS

|  |  |  |
| --- | --- | --- |
| Класс | Решилось без случайного поиска | Решилось со случайным поиском |
| GKLS\_nd\_10 | 78 | 99 |
| GKLS\_nd\_30 | 0 | 15 |
| GKLS\_cd\_10 | 67 | 96 |
| GKLS\_cd\_30 | 0 | 1 |
| GKLS\_td\_10 | 65 | 94 |
| GKLS\_td\_30 | 0 | 0 |

Ниже приведены результаты численного сравнения двух последовательных алгоритмов – DIRECT [] и алгоритма глобального поиска (АГП без случайного поиска). Численное сравнение проводилось на классах функций GKLS\_nd, GKLS\_cd и GKLS\_td размерности 10. Максимально допустимое число итераций составляло *Kmax = 1 000 000*. В таблице 4 отражено среднее число итераций и количество решенных задач данного класса.

Таблица 4. Использование случайного поиска при решении задач классов GKLS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Класс | Direct | | | АГП | |
| Решилось задач | | Среднее число итераций | Решилось задач | Среднее число итераций |
| GKLS\_nd | 42 | 182439 | | 78 | 118377 |
| GKLS\_cd | 43 | 397291 | | 67 | 269450 |
| GKLS\_td | 45 | 450101 | | 65 | 351866 |

Как видно из таблицы 4, АГП превосходит метод DIRECT на рассмотренных классах задач как по среднему числу итераций так и по числу решенных задач.

Оценим теперь ускорение, которое достигается при использовании параллельного алгоритма глобального поиска, в зависимости от числа *P* используемых ядер. Численное сравнение проводилось на классах функций GKLS\_nd, GKLS\_cd и GKLS\_td размерности 10. Максимально допустимое число итераций составляло *Kmax* = 1 000 000. В таблице 5 приведено число решившихся задач, в таблице 6 – среднее время решения одной задачи, а в таблице 7 – ускорение по времени относительно последовательного запуска.

Таблица 5. Количество решенных задач по классам, в зависимости от числа потоков

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Problem class | P = 2 | P = 4 | P = 8 | P = 16 |
| GKLS\_nd\_10 | 81 | 88 | 84 | 82 |
| GKLS\_cd\_10 |  |  |  |  |
| GKLS\_td\_10 |  |  |  |  |

Таблица 6. Среднее время решения задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Problem class | P = 2 | P = 4 | P = 8 | P = 16 |
| GKLS\_nd\_10 | 15,80 | 16,30 | 15,01 | 12,55 |
| GKLS\_cd\_10 |  |  |  |  |
| GKLS\_td\_10 |  |  |  |  |

Таблица 7. Ускорение по времени

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Problem class | P = 2 | P = 4 | P = 8 | P = 16 |
| GKLS\_nd\_10 | 1,48 | 1,43 | 1,56 | 1,86 |
| GKLS\_cd\_10 |  |  |  |  |
| GKLS\_td\_10 |  |  |  |  |

Результаты экспериментов показывают преимущество параллельного алгоритма по сравнению с последовательным и по числу решенных задач, и по времени решения. Но поскольку для десятимерных задач необходимо использовать расширенную точность при вычислениях, то параллельное проведение испытаний не дает значительного ускорения.

Далее приведены результаты для многопроцессорного запуска. Численное сравнение проводилось на классах функций GKLS\_nd, GKLS\_cd и GKLS\_td размерности 10. В соответствии с блочной рекурсивной схемой было использовано два уровня подзадач с размерностями . В таблице 8 приведено число решившихся задач, среднее время решения одной задачи и ускорение по времени относительно последовательного запуска.

Таблица 8. табл

|  |  |  |  |
| --- | --- | --- | --- |
| Problem class | Решилось задач | среднее время | ускорение |
| GKLS\_nd\_10 | 2 | 1,044186 | 22,39532469 |
| GKLS\_cd\_10 | 1 | 0,038169 | 50,92690803 |
| GKLS\_td\_10 | 2 | 0,0356465 | 66,46523436 |

# 5. Заключение

В работе представлены результаты решения 10-ти и 30-ти мерных задач безусловной глобальной оптимизации с конкурса GenOpt 2016. Описаны использованные методы оптимизации и их модификации, направленные на получение решения при заданном ограничении в один миллион испытаний. Все рассмотренные модификации реализованы в решателе ExaMin, использованном при проведении экспериментов.

# Литература

1. 1. Hill, J.D.: A search technique for multimodal surfaces. IEEE Transactions on Systems Science and Cybernetics. 5(1), 2{8 (1969)
2. 2. Shekel J.: Test functions for multimodal search technique. Proceedings of the 5th Princeton Conference on Information Science Systems. Princeton, Princeton University Press. 354{359 (1971)
3. 4. Sergeyev, Ya.D., Grishagin, V.A.: Sequential and parallel global optimization algorithms. Optimization Methods and Software. 3, 111--124 (1994)
4. 5. Gergel, V.P.: A method of using derivatives in the minimization of multiextremum functions. Computational Mathematics and Mathematical Physics 36(6), 729--742 (1996)
5. 6. Gergel, V.P.: A global optimization algorithm for multivariate functions with lipschitzian first derivatives. J. Glob. Optim.10(3), 257--281 (1997)
6. 9. Sergeyev Ya.D, Grishagin V.A.: Parallel asynchronous global search and the nested optimization scheme. Journal of Computational Analysis and Applications. 3(2), 123--145 (2001)
7. 10. Barkalov, K.A., Strongin, R.G.: A global optimization technique with an adaptive order of checking for constraints. Computational Mathematics and Mathematical Physics 42(9), 1289--1300 (2002)
8. 11. Strongin, R.G., Sergeyev, Ya.D.: Global optimization: fractal approach and non-redundant parallelism. J. Glob. Optim. 27(1), 25--50 (2003)
9. 10. Sergeyev, Ya.D., Grishagin, V.A.: A parallel method for finding the global minimum of univariate functions. J. Optim. Theory Appl. 80(3), 513{536 (1994)
10. 11. Sergeyev, Ya.D., Grishagin, V.A.: Sequential and parallel global optimization algorithms. Optimization Methods and Software. 3, 111{124 (1994)
11. 12. Gergel, V.P.: A method of using derivatives in the minimization of multiextremum functions. Computational Mathematics and Mathematical Physics 36(6), 729{742 (1996)
12. 13. Gergel, V.P.: A global optimization algorithm for multivariate functions with lipschitzian first derivatives. J. Glob. Optim. 10(3), 257{281 (1997)
13. 14. Grishagin, V.A., Sergeyev, Ya.D., Strongin, R.G.: Parallel characteristical algorithms for solving problems of global optimization. J. Glob. Optim. 10(2), 185{206(1997)
14. 15. Gergel, V.P., Sergeyev, Ya.D.: Sequential and parallel algorithms for global minimizing functions with Lipschitzian derivatives. Computers and Mathematics with Applications, 37(4-5), 163{179 (1999)
15. 16. Sergeyev Ya.D, Grishagin V.A.: Parallel asynchronous global search and the nested optimization scheme. Journal of Computational Analysis and Applications. 3(2), 123{145 (2001)
16. 17. Strongin, R.G., Sergeyev, Ya.D.: Global optimization: fractal approach and non redundant parallelism. J. Glob. Optim. 27(1), 25{50 (2003)
17. 18. Gergel, V.P., Strongin, R.G.: Parallel computing for globally optimal decision making on cluster systems. Future Generation Computer Systems, 21(5), 673{678 (2005)
18. 19. Barkalov, K., Polovinkin, A., Meyerov, I., Sidorov, S., Zolotykh, N.: SVM regression parameters optimization using parallel global search algorithm. Lecture Notes in Computer Science. 7979, 154{166 (2013)
19. 20. Barkalov, K.A., Gergel, V.P.: Multilevel scheme of dimensionality reduction for parallel global search algorithms. Proceedings of the 1st International Conference on Engineering and Applied Sciences Optimization { OPT-i 2014. pp. 2111{2124. (2014)
20. 21. Gergel, V., Grishagin, V., Israfilov, R.: Local tuning in nested scheme of global optimization. Procedia Computer Science, 51(1), pp. 865{874. (2015)
21. 22. Gergel, V., Grishagin, V., Gergel, A.: Adaptive nested optimization scheme for multidimensional global search. Journal of Global Optimization, 17 p. Article in Press. (2015)
22. 23. Barkalov, K., Gergel, V.: Parallel global optimization on GPU. Journal of Global Optimization. 18 p. Article in Press. (2016)
23. Сергеев Я.Д., Квасов Д.Е. Диагональные методы глобальной оптимизации. – М.: Физматлит, 2008. -352 c.
24. Gaviano, M. Software for generation of classes of test functions with known local and global minima for global optimization/ M. Gaviano, D. Lera, D. E. Kvasov, Y. D. Sergeyev // ACM Transactions on Mathematical Software. – 2003. – Vol. 29. – P. 469-480.
25. R.G. Strongin, Ya.D. Sergeyev, Global optimization with non-convex constraints. Sequential and parallel algorithms. Kluwer Academic Publishers, Dordrecht, 2000.
26. Стронгин Р.Г., Гергель В.П., Гришагин В.А., Баркалов К.А. Параллельные вычисления в задачах глобальной оптимизации. М.: Издательство Московского университета. 2013. 280 с.
27. Городецкий С.Ю., Гришагин В.А. Нелинейное программирование и многоэкстремальная оптимизация. Н.Новгород: Изд-во ННГУ, 2007.
28. Hooke, R., Jeeves, T.A.: "Direct search" solution of numerical and statistical problems // Journal of the ACM. 8(2), 212{229 (1961)
29. 27. Wilde, D.J.: Optimum Seeking Methods. Prentice-Hall, Engelwood Cliffs, NewJersey (1964)
30. 28. Himmelblau, D.M.: Applied Nonlinear Programming. McGraw-Hill, New York (1972)

Use of parallel ExaMin solver for solving global optimization problems from GenOpt competition

V.V. Sovrasov

Lobachevsky State University of Nizhni Novgorod

In this work parallel algorithms for solving multiextremal optimization problems are considered. These algorithms have been developed in the information-statistical approach proposed in Lobachevsky State University of Nizhny Novgorod and have been implemented in parallel software ExaMin. Results of numerical experiments carried out with test function from international competition of optimization algorithms GenOpt (www.genopt.org ) are discussed.

*Keywords:* global optimization, multiextremal functions, dimension reduction, parallel algorithms

# References

1. Eremin I.I. Fejerovskiemetodydlyazadachlinejnojivypuklojoptimizatsii [Fejer Methods for Problems of Convex and Linear Optimization]. Chelyabinsk, Publishing of the South Ural State University, 2009. 200 p.
2. Levin V.K. Otechestvennyesuperkomputerysemejstva MVS [National Family of MVS Supercomputers]. URL: http://parallel.ru/mvs/levin.html (accessed: 27.05.2012).
3. Akimova E.N., Belousov D.V. Rasparallelivanieresheniyalineynoyobratnoyzadachina MVS-1000 igraficheskikhprotsessorakh [Parallelization of linear inverse problem on the MVS-1000 and GPUs]. Parallelnyevychislitelnyetekhnologii (PaVT'2010): Trudy mezhdunarodnojnauchnojkonferentsii (Ufa, 29 marta – 2 aprelya 2010) [Parallel Computational Technologies (PCT'2010): Proceedings of the International Scientific Conference (Ufa, Russia, March, 29 – April, 2, 2010)]. Chelyabinsk, Publishing of the South Ural State University, 2010. P. 18–27.
4. Sokolinsky L.B. Organization of Parallel Query Processing in Multiprocessor Database Machines with Hierarchical Architecture // Programming and Computer Software. 2001. Vol. 27, No. 6. P. 297–308.
5. Amit Y. 2D Object Detection and Recognition: models, algorithms and networks. MIT Press, 2002. 325 p.
6. Cadez I., Heckerman D., Meek C., et al. Visualization of Navigation Patterns on a Web Site Using Model Based Clustering. Technical Report MSR‑TR‑00‑18. Microsoft Research. 2000. URL: http://research.microsoft.com/pubs/69752/tr-2000-18.pdf (accessed: 13.12.2009).
7. Levshin D.V., Markov A.S. Algorithms for Integrating PostgreSQL with the Semantic Web // Programming and Computer Software. 2009. Vol. 35, No. 3. P. 136–144.
8. Stonebraker M., Kemnitz G. The POSTGRES Next-generation Database Management System // Communications of the ACM. 1991. Vol. 34, No. 10. P. 78–92.

1. \* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 16-31-00244 мол\_а «Параллельные методы решения вычислительно трудоемких задач глобальной оптимизации на гибридных кластерных системах». [↑](#footnote-ref-2)