

ПАРАЛЛЕЛЬНЫЙ ПОИСК ГЛОБАЛЬНОГО ЭКСТРЕМУМА ФУНКЦИЙ МНОГИХ ПЕРЕМЕННЫХ¹

© 2008 г. Ю. Г. Евтушенко, В. У. Малкова, А. А. Станевичюс

(119991 Москва, ул. Вавилова, 40, ВЦ РАН)

e-mail: evt@ccas.ru; malkova@ccas.ru; stanev@ccas.ru

Поступила в редакцию 03.06.2008 г.

На основе метода неравномерных покрытий разработан метод параллельного поиска глобального экстремума липшицевых функций. На языке C в системе параллельного программирования реализован метод неравномерных покрытий для глобальной минимизации функций, градиент которых удовлетворяют условию Липшица. Работа алгоритма демонстрируется на примере расчета строения молекулы протеина.

Ключевые слова: глобальный экстремум, метод неравномерных покрытий, липшицевы функции, параллельные вычисления, многопроцессорные системы, функция Морса энергии атомного кластера

1. Введение

В подавляющем большинстве оптимизационных задач требуется найти глобальный оптимум. Однако из-за большой сложности поиска глобального экстремума обычно ограничиваются отысканием локальных решений. Современные многопроцессорные ЭВМ существенно расширяют возможности решения задач глобальной оптимизации, позволяя использовать параллельные алгоритмы расчетов. Обзор многочисленных публикаций по поиску глобального экстремума приведен в книге [1]. Проблемы распараллеливания вычислений подробно изложены в [2].

В 1971 г. в [3] был предложен и на языке АЛГОЛ-60 реализован метод неравномерных покрытий для нахождения глобального экстремума липшицевых функций. Дальнейшее развитие этот метод получил в [4], [5]. Если известна константа Липшица, то метод неравномерных покрытий дает гарантированное решение, но на практике им удается решать лишь задачи невысокой размерности. Численные эксперименты показывают, что эффективность метода существенно зависит от константы Липшица, которая обычно априори не известна и приходится пользоваться ее завышенными верхними оценками, что существенно усложняет расчеты. Вместе с тем, во многих задачах существуют подобласти, где константа Липшица невелика и ориентация на максимальную константу не оправдана. Ниже рассмотрена глобальная минимизация трех типов функций:

1. функции, удовлетворяющие условию Липшица,
2. функции, градиент которых удовлетворяет условию Липшица,

¹Работа выполнена при финансовой поддержке программы № 14 фундаментальных исследований Президиума РАН "Раздел II: Высокопроизводительные вычисления и многопроцессорные системы" 2007 г., программы № 15 Президиума РАН "Исследование и создание научных приложений в распределенной среде суперкомпьютерных вычислений на базе ВЦ РАН" 2007 г., а также гранта Президента РФ НШ-5073.2008.1)

3. функции, гессиан которых удовлетворяет условию Липшица.

Для всех этих типов разработана общая схема неравномерных покрытий с использованием программ локальной оптимизации.

Параллельная реализация метода неравномерных покрытий и некоторые результаты расчетов для функции, удовлетворяющей условию Липшица, описаны в [6]. В отличие от [6], в данной работе основное внимание уделяется глобальной минимизации функций, градиенты или гессианы которых удовлетворяют условию Липшица. На языке C в системе параллельного программирования MPI (Message Passing Interface) разработана программа, реализующая поиск глобального экстремума функций, градиенты которых удовлетворяют условию Липшица.

2. Постановка задачи, общая идея метода покрытий

Рассмотрим задачу отыскания глобального минимума функции f на множестве $X \subset \mathbb{R}^n$. Обозначим

$$f_* = \text{glob min}_{x \in X} f(x). \quad (1)$$

Через X_* обозначим множество решений задачи (1). Введем множество ε -оптимальных (приближенных) решений задачи (1):

$$X_*^\varepsilon = \{x \in X : f(x) \leq f_* + \varepsilon\}. \quad (2)$$

Очевидно, что $X_* \subset X_*^\varepsilon \subset X$. В большинстве практических задач достаточно найти хотя бы одну точку $x_r \in X_*^\varepsilon$ и взять в качестве оптимального значения f_* число $f_r = f(x_r)$. Другими словами, надо с заданной точностью ε определить величину глобального минимума функции и найти хотя бы одну точку x_r , где это приближенное значение достигается.

Введем набор n -мерных точек $N_k = [x_1, \dots, x_k]$, где все $x_i \in X$. В каждой точке x_i вычисляется значение функции f , и по формуле

$$R_i = \min_{1 \leq j \leq i} f(x_j) = f(x_\alpha), 1 \leq \alpha \leq i \quad (3)$$

определяется текущий рекорд R_i и одна из рекордных точек $x_\alpha, 1 \leq \alpha \leq i$. Из (3) следует, что последовательность рекордов R_i невозрастающая.

С каждой точкой x_i свяжем некоторую ее окрестность P_i , при этом $x_i \in P_i$. В качестве P_i можно взять, например, n -мерный параллелепипед, содержащий x_i , или шар с центром в точке x_i .

Введем множества X_i и определим их объединение

$$X_i = \{x : x_i \in P_i, f(x) \geq R_i - \varepsilon\}, 1 \leq i \leq k, \quad V_k = \bigcup_{i=1}^k X_i. \quad (4)$$

Будем говорить, что набор множеств $S_k = [X_1, \dots, X_k]$ покрывает множество X , если

$$X \subset V_k \quad (5)$$

Теорема 1. Пусть набор допустимых точек N_k и соответствующих множеств S_k таковы, что выполнено условие (5); тогда определяемая из условий $R_k = f(x_r), x_r \in N_k$ рекордная точка x_r принадлежит множеству X_^ε .*

Действительно, если набор множеств S_k полностью покрывает X , то каждая точка x_* из X_* будет принадлежать по крайней мере одному из множеств X_i . Пусть $x_* \in X_s, s \leq k$, тогда $R_s \geq R_k$ и, учитывая (3), получим

$$f_* = f(x_*) \geq R_s - \varepsilon \geq R_k - \varepsilon = f(x_r) - \varepsilon.$$

Согласно (2) отсюда следует, что $x_r \in X_*^\varepsilon$.

Данная теорема выражает основную идею метода неравномерных покрытий: ищутся глобальные минимумы на подмножествах, объединение которых содержит X , вместо поиска глобального минимума на X . Условие (4) можно проверить, построив для функции $f(x)$ ее миноранту на множествах P_i или X_i . Миноранты функций, удовлетворяющих условию Липшица, даны в следующем параграфе.

При реализации метода наборы S_k и N_k строятся последовательно, после каждого вычисления значения функции f происходит уточнение рекорда, поэтому $R_i \leq f(x_i)$ при всех i . Считаем, что последнее вычисление f было выполнено в точке c_m , рекордная точка была $c_r, r \leq m$, а рекорд был R_m . Определим следующее лебегово множество

$$\mathcal{L}_m = \{x \in P : R_m - \varepsilon \leq f(x)\}. \quad (6)$$

Пусть глобальный минимум функции $f(x)$ на \mathcal{L}_m достигается в точке x_l . Тогда $R_m - f(x_l) = f(c_r) - f(x_l) \leq \varepsilon$ и, следовательно, в результате глобальной минимизации на \mathcal{L}_m рекорд R_m может быть уточнен не более чем на ε . Поэтому множество \mathcal{L}_m не представляет интереса и может быть исключено из области поиска X .

Если после описанных вычислений условие $X \subset V_m$ не выполнено, то поиск минимума продолжается на множестве $W_m = X \setminus V_m$. Пусть определены $c_{m+1} \in X_{m+1} \subseteq W_m$, вычислены $f(c_{m+1})$ и R_{m+1} . Если на X_{m+1} выполнено условие $f(x) \geq R_{m+1} - \varepsilon$, то множество X_{m+1} может быть исключено и поиск минимума продолжается на $W_m \setminus X_{m+1}$. В противном случае множество X_{m+1} разбивается на несколько подмножеств, в каждом из них вычисляются значения функции f , пересчитывается рекорд, некоторые подмножества исключаются, другие дробятся и т.д. Процесс заканчивается, когда все множество X будет полностью покрыто. Последовательность рекордов – невозрастающая. Последовательность лебеговых множеств \mathcal{L}_m , напротив, расширяющаяся, т.е. $\mathcal{L}_m \subseteq \mathcal{L}_{m+i}$ при $i \geq 0$.

Лебегово множество \mathcal{L}_m будет наибольшим, если в (6) рекорд $R_m = f_*$. Однако априори величина f_* не известна. Поэтому для уменьшения рекорда целесообразно использовать методы поиска локального минимума функции f на множестве X . В тех точках $c_i \in X$, где получено, что $f(c_i) < R_m$, происходит улучшение рекорда, и есть основание полагать, что решая задачу локального поиска

$$\operatorname{loc} \min_{x \in X} f(x),$$

стартуя из точки c_i , удастся найти другую точку $\bar{c} \in X$, в которой $f(\bar{c}) < f(c_i)$. Такой прием часто существенно ускоряет расчеты, позволяя улучшить рекорд и тем самым расширить область, которую можно отбрасывать в процессе покрытия множества X .

3. Поиск минимума липшицевых функций

Наиболее трудным этапом в реализации метода неравномерных покрытий является определение хотя бы некоторой части множества \mathcal{L}_m . Здесь приходится налагать на функцию f дополнительные требования. Предположим, что для $f(x)$ можно построить мажоранту $G(x, y)$ и миноранту $g(x, y)$ так, что для всех $x, y \in X$ выполнены неравенства

$$G(x, y) \geq f(x) \geq g(x, y), G(y, y) = g(y, y) = f(y). \quad (7)$$

Определим множества

$$K_i = \{x \in P_i : g(x, c_i) \geq R_i - \varepsilon\}, U_k = \bigcup_{i=1}^k K_i. \quad (8)$$

Из (6) и (7) следует, что $K_i \subseteq \mathcal{L}_m$, поэтому множество K_i можно исключить из области поиска X . Сравнивая определения (4) и (8), получим что $K_i \subset X_i, U_k \subset V_k$. Теорему 1 можно переформулировать так: если набор допустимых точек N_k и соответствующие множества K_1, \dots, K_k таковы, что $X \subset U_k$, то определяемая из условий $R_k = f(x_r), x_r \in N_k$ рекордная точка x_r принадлежит X_*^ε .

Для конструктивной реализации покрытия множества X необходимо уметь строить миноранты. Их нахождение особенно просто для липшицевых функций. Рассмотрим три класса таких функций.

1. Пусть функция f удовлетворяет условию Липшица с константой l , т.е. для любых x и y из X выполнено неравенство

$$|f(x) - f(y)| \leq l \|x - y\|. \quad (9)$$

Отсюда получим мажоранту $G(x, y)$ и миноранту $g(x, y)$:

$$G(x, y) = f(y) + l \|x - y\| \geq f(x) \geq f(y) - l \|x - y\| = g(x, y). \quad (10)$$

Используя эту миноранту, определение (8) запишем в виде

$$K_i = P_i \cap H_i, H_i = \{x \in R^n : \|x - c_i\| \leq \rho_{ir}\}, \quad (11)$$

где

$$\rho_{ir} = (\varepsilon + f(c_i) - R_r)/l, 1 \leq r \leq i \leq k. \quad (12)$$

Если $\|\cdot\|$ – евклидова норма, то H_i – шар с радиусом ρ_{ir} и центром в точке c_i ; если $\|\cdot\|$ – чебышевская норма, то H_i – куб с центром в точке c_i и главной диагональю, равной $2\rho_{ir}$.

Векторы c_i принадлежат X для всех $1 \leq i \leq k$. Согласно (3) $f(c_i) \geq f(c_r) = R_r, 1 \leq r \leq i$. Поэтому радиус шара ρ_{ir} , задаваемый формулой (12), будет всегда больше или равен ε/l . Обозначим

$$\xi = \frac{2\varepsilon}{l}. \quad (13)$$

Величину ξ будем называть минимальным диаметром покрывающего шара. Такой диаметр реализуется, если $f(c_i) = R_r$. Для $x \in H_i$ имеем:

$$f(x) \geq \min_{x \in H_i} g(x, c_i) = f(c_i) - l \max_{x \in H_i} \|x - c_i\| = f(c_i) - l\rho_{ir} \geq R_r - \varepsilon.$$

Следовательно, шар H_i можно исключить из области поиска X . Если объединение шаров $H_i, 1 \leq i \leq k$, удовлетворяющих (11), покрыло множество X , то $c_r \in X_*^\varepsilon$.

Радиус шара H_i будет большим, если $f(c_i) \gg f(c_r)$, и малым в областях, где $f(c_i) \approx f(c_r)$. Благодаря этому возможно неравномерное покрытие множества X шарами различного радиуса. Впервые эта идея была высказана и реализована в [3].

2. Пусть функция $f(x)$ дифференцируема и ее градиент удовлетворяет условию Липшица, т.е. существует константа L такая, что для любых x и y из X имеет место неравенство

$$\|f_x(x) - f_x(y)\| \leq L\|x - y\| \quad (14)$$

Несложно показать, что если $f(x)$ дважды дифференцируема, то (14) имеет место тогда и только тогда, когда $\|f_{xx}(x)\| \leq L$ при любых $x \in X$. Из (14) следует, что вогнутая опорная миноранта $g(x, y)$ и выпуклая опорная мажоранта $G(x, y)$ функции $f(x)$ определяются по формулам

$$g_1(x, y) = f(y) + \langle f_x(y), x - y \rangle - \frac{L}{2}\|x - y\|^2, \quad (15)$$

$$G_1(x, y) = f(y) + \langle f_x(y), x - y \rangle + \frac{L}{2}\|x - y\|^2. \quad (16)$$

Отсюда следует неравенство

$$L\|x - y\|^2 \geq 2|f(x) - f(y) - \langle f_x(y), x - y \rangle|. \quad (17)$$

Эту формулу можно использовать для проверки правильности выбора константы Липшица L . Невыполнение условия (17) указывает на то, что либо величина L должна быть увеличена, либо предположение (14) не имеет места для данной функции, либо условия (15) и (16) можно использовать лишь локально, на некоторых подмножествах $H_i \subset X$.

Применим формулы (15) и (16) для нахождения глобального минимума функции f на множестве X . Будем искать минимум миноранты и мажоранты по x . При $y \in X$ определим векторы \hat{x}_1 и \hat{x}_2 такие, что

$$\hat{x}_1 \in \arg \min_{x \in X} [\langle f_x(y), x - y \rangle - \frac{L}{2}\|x - y\|^2], \quad (18)$$

$$\hat{x}_2 \in \arg \min_{x \in X} G_1(x, y). \quad (19)$$

Вторая задача много проще, т.к. $G_1(x, y)$ – выпуклая функция. Очевидно, что

$$f(\hat{x}_1) \leq \min_{x \in X} f(x) \leq f(\hat{x}_2). \quad (20)$$

Если \hat{x}_2 – внутренняя точка множества X , то

$$\hat{x}_2 = y - \frac{f_x(y)}{L}, f(\hat{x}_2) = f(y) - \frac{\|f_x(y)\|^2}{2L}$$

При численной реализации метода отыскивался минимум функции $f(x)$ на отрезках, соединяющих точки y и \hat{x}_1 и точки y и \hat{x}_2 . Во многих случаях эта процедура позволяла улучшить рекорд.

Из (10) и (15) следует, что для всех x и y из X имеет место неравенство

$$f(x) \geq f(y) + \langle f_x(y), x - y \rangle - \frac{L}{2} \|x - y\|^2. \quad (21)$$

Условие $f(x) \geq R_r - \varepsilon$ выполнено, если x удовлетворяет условию

$$\frac{L\|x - y\|^2}{2} + \langle f_x(y), x - y \rangle \leq \varepsilon + f(y) - R_r. \quad (22)$$

Это множество можно удалить из X при численном поиске глобального минимума.

В случае, когда X – параллелепипед, задача о минимизации миноранты имеет аналитическое решение, приведенное в следующем разделе.

3. Пусть f – дважды дифференцируемая функция с гессианом, удовлетворяющим условию Липшица с константой M :

$$\|f_{xx}(x) - f_{xx}(y)\| \leq M\|x - y\|. \quad (23)$$

Согласно (14) опорная миноранта и мажоранта функции $f(x)$ в этом случае имеют вид

$$g_2(x, y) = f(y) + \langle f_x(y), x - y \rangle + \frac{1}{2}(x - y)^T f_{xx}(y)(x - y) - \frac{M}{6} \|x - y\|^3$$

$$G_2(x, y) = f(y) + \langle f_x(y), x - y \rangle + \frac{1}{2}(x - y)^T f_{xx}(y)(x - y) + \frac{M}{6} \|x - y\|^3$$

В отличие от (15) и (16) эти формулы дают аппроксимации второго порядка точности для функции $f(x)$. Вообще говоря, здесь обе функции не являются выпуклыми. Минимизации мажоранты $G_2(x, y)$ посвящена работа [7].

Пусть $\gamma(y)$ – наименьшее собственное значение матрицы $f_{xx}(y)$. Обозначим $\Delta = \|x - y\|$, тогда

$$g_2(x, y) \geq f(y) + [-\|f_x(y)\| + \frac{1}{2}\gamma(y)\Delta - \frac{M}{6}\Delta^2]\Delta.$$

Условие $f(x) \geq R_r - \varepsilon$ выполнено внутри шара с центром, расположенном в точке y , и радиусом Δ , удовлетворяющем условию

$$\Delta^3 + \frac{3\Delta}{M}[\gamma(y)\Delta - 2\|f_x(y)\|] \leq \frac{6}{M}[\varepsilon + f(y) - R_r], \quad (24)$$

где $f(y) \geq R_r$.

Симметричную матрицу $f_{xx}(y)$ представим в виде разности двух матриц

$$f_{xx}(y) = f_{xx}^1(y) - f_{xx}^2(y),$$

где $f_{xx}^1(y)$ и $f_{xx}^2(y)$ – положительно определенные матрицы. Миноранта $g_2(x, y)$ приведена к д.с. виду, т.е. к разности двух выпуклых функций:

$$g_2(x, y) = g_{21}(x, y) - g_{22}(x, y),$$

где

$$g_{21}(x, y) = f(y) + \langle f_x(y), x - y \rangle + \frac{1}{2}(x - y)f_{xx}^1(y)(x - y),$$

$$g_{22}(x, y) = \frac{1}{2}((x - y)^T f_{xx}^2(y)(x - y) + \frac{M}{6}\|x - y\|^3).$$

Обширную библиографию по д.с. функциям, методам их оптимизации можно найти, например, в [8].

Излагаемый подход можно продолжить и рассмотреть случай, когда j -ая производная функции f удовлетворяет условию Липшица. С увеличением j происходит более точная аппроксимация мажоранты и миноранты функции f , что повышает эффективность метода неравномерных покрытий, однако одновременно возрастает сложность расчетов, связанная с необходимостью находить производные функции $f(x)$ высоких порядков.

Расчеты по предлагаемым схемам гарантируют отыскание глобального минимума, но при том условии, что известны константы Липшица. Основной недостаток такого подхода состоит в том, что в качестве констант Липшица приходится брать некоторые их верхние оценки, а это существенно усложняет расчеты, особенно в тех подобластях, где градиент функции $f(x)$ мал. Простейший выход из положения состоит в введении предположения, что в каждом i -м шаре H_i есть своя константа Липшица. Для этого (9) и (14) заменяются условиями

$$l_i = \max_{y \in H_i} \max_{x \in H_i} |f(x) - f(y)|/\Delta,$$

$$L_i = \max_{y \in H_i} \max_{x \in H_i} |f_x(x) - f_x(y)|/\Delta.$$

Нахождение точных констант по этим формулам может оказаться сложней поиска глобального минимума $f(x)$ на H_i . Поэтому можно ограничиться приближенными оценками. В частности, если диаметр множества H_i достаточно мал, то можно полагать $l_i \simeq \|f_x(c_i)\|$, $L_i \simeq \|f_{xx}(c_i)\|$, где c_i – центр H_i .

4. Алгоритм последовательного покрытия

При программной реализации метода вводятся дополнительные упрощающие предположения. Считается, что допустимое множество X – это n -мерный параллелепипед с гранями, параллельными координатным плоскостям:

$$X = \{x \in R^n, a \leq x \leq b\}.$$

Здесь и ниже неравенство $a \leq x$ означает, что $a^j \leq x^j$ для всех $1 \leq j \leq n$.

В процессе расчетов используются вспомогательные векторы $a_i, b_i \in R^n$ и порождаемые ими прямоугольные параллелепипеды с гранями, параллельными координатным плоскостям: $P_i = \{x \in R^n : a_i \leq x \leq b_i\}$.

Будем считать, что $a \leq a_i < b_i \leq b$. Таким образом, все параллелепипеды $P_i \subseteq X$. В качестве вектора y берется $c_i = (a_i + b_i)/2$ – центр параллелепипеда P_i . Тогда, радиус i -го параллелепипеда,

$$\Delta_i = \|c_i - a_i\| = \|c_i - b_i\| = \frac{1}{2}\|a_i - b_i\|.$$

Чебышевская норма вектора $d_i = b_i - a_i$ определяет длину максимального ребра w_i параллелепипеда P_i :

$$w_i = \|d_i\|_\infty = \max_{1 \leq j \leq n} d_i^j = 2\Delta_i.$$

Через q_i обозначим значение минимума миноранты функции $f(x)$ на P_i . Если $f(x)$ удовлетворяет условию Липшица (9) всюду на P_i с константой l_i , то согласно (10)

$$q_i = \min_{x \in P_i} g(c_i, x) = f(c_i) - l_i \Delta_i. \quad (25)$$

Если градиент функции $f(x)$ удовлетворяет условию (14) всюду на P_i с константой L_i , то q_i определяется из минимизации миноранты (15) на P_i , то есть

$$q_i = \min_{x \in P_i} [f(c_i) + \langle f_x(c_i), x - c_i \rangle - \frac{L_i}{2} \|x - c_i\|^2].$$

При всех $x \in P_i$ имеют место неравенства

$$a_i \leq x_i \leq b_i, \frac{(a_i - b_i)}{2} \leq x_i - c_i \leq \frac{(b_i - a_i)}{2}.$$

Минимум миноранты по x_i^j на P_i достигается на границе при $x_*^j = b_i^j$, если $\frac{\partial f(c_i)}{\partial x^j} < 0$, и при $x_*^j = a_i^j$ в противном случае. В обоих случаях $|x_*^j - c_i^j| = b_i^j - c_i^j, 1 \leq j \leq n$,

$$q_i = f(c_i) - \langle |f_x(c_i)|, b_i - c_i \rangle - \frac{L_i}{2} \|b_i - c_i\|^2.$$

Здесь $|f_x(c_i)|$ обозначает n -мерный вектор, у которого каждая j -ая компонента есть $|\frac{\partial f(c_i)}{\partial x^j}|$. Отсюда следует нижняя оценка минимума миноранты на параллелепипеде P_i

$$q_i \geq f(c_i) - Q_i \|b_i - c_i\| - \frac{L_i}{2} \|b_i - c_i\|^2, \text{ где } Q_i = \|f_x(c_i)\|. \quad (26)$$

Если $f(x_*) < R_r$, то положим $R_r = f(x_*)$. Согласно (26) неравенства

$$f(x) \geq q_i \geq R_r - \varepsilon \quad (27)$$

имеют место, если

$$\|b_i - c_i\|^2 + \frac{2}{L_i} Q_i \|b_i - c_i\| \leq \varphi_{ir}, \text{ где } \varphi_{ir} = \frac{2}{L_i} (\varepsilon + f(c_i) - R_r) \geq \frac{2\varepsilon}{L_i}.$$

Введем шар S_i с радиусом ζ_{ir}

$$S_i = \{x : \|x - c_i\| \leq \zeta_{ir}\},$$

где

$$\zeta_{ir} = \sqrt{\theta_i^2 + \varphi_{ir}} - \theta_i, \text{ где } \theta_i = \frac{Q_i}{L_i}.$$

При $Q_i = 0$ имеем $\zeta_{ir} = \sqrt{\varphi_{ir}}$, диаметр шара будет минимальным (по возможным значениям $f(c_i) \geq R_r$) и равным

$$\xi = 2\sqrt{\frac{2\varepsilon}{L_i}}. \quad (28)$$

При $L_i = 0$ получим $\zeta_{ir} = \varepsilon + f(c_i) - R_r \geq \varepsilon$.

При выполнении условий (9), (14), минимальный диаметр покрывающего шара определяется из (13) и (28) соответственно. Если выполнено условие (23), то ξ находится из (24).

Если

$$\Delta_i \leq \zeta_{ir}, \quad (29)$$

то $P_i \subset S_i$ и параллелепипед P_i можно исключить из P , продолжая поиск минимума на множестве $P \setminus S_i$. Если неравенство (29) не выполняется, то производится последовательное дробление параллелепипеда P_i по какому-нибудь ребру.

Сечение P_i по ребру максимальной длины представляется наиболее простым и естественным. Однако в случае большого разброса значений компонент вектора градиента можно ввести некоторый аналог масштабирования. Если функция $f(x)$ дифференцируема, то в качестве ребра, подлежащего сечению, можно взять ребро под номером α , для которого

$$\alpha = \arg \max_{1 \leq j \leq n} (b_i^j - c_i^j) \left| \frac{\partial f(c_i)}{\partial x^j} \right|. \quad (30)$$

Такой вариант сечений предложен в [9]. В частном случае, если все компоненты вектора градиента одинаковы, то α будет номером наибольшего ребра.

Если не использовать дифференцируемость, то условие выбора (30) можно заменить следующим:

$$\alpha = \arg \max_{1 \leq j \leq n} |f(c_{i2}^j) - f(c_{i1}^j)|, \quad (31)$$

где векторы c_{i2}^j и c_{i1}^j имеют те же компоненты, что и вектор c_i за исключением j -х компонент, которые у векторов c_{i2}^j и c_{i1}^j есть a_i^j и b_i^j соответственно.

Программная реализация метода неравномерных покрытий выполнена однотипным образом для разных липшицевых функций. По-разному вычисляются только оценки минимума q_i и минимальные диаметры покрывающих шаров. Алгоритм покрытия множества X параллелепипедами P_i остается тот же.

Если $q_i \geq R_i - \varepsilon$, то параллелепипед P_i может быть исключен, так как глобальный минимум на нем не дает улучшения текущего рекорда R_m более чем на ε , и поиск продолжается на множестве $X \setminus P_i$. В противном случае, если $q_i < R_m - \varepsilon$, то параллелепипед P_i делится пополам, например, по максимальному ребру и поиск ведется на этих двух частях. Если главные диагонали одной или обеих частей меньше, чем ξ , то на них $f(x) \geq q_i = f(c_i) - \varepsilon$ и эти части исключаются из области поиска. В центрах оставшихся частей вычисляются значения $f(x)$. При этом, если возможно, улучшается рекорд и проверяется условие покрытия обеих частей. Те части, которые оказались покрытыми, исключаются из области поиска. Если обе

части исключены, то P_i также исключается, если нет, то продолжается дальнейшее деление до тех пор, пока параллелепипед P_i не будет покрыт. Возможны самые разнообразные способы покрытия параллелепипеда X . В [3], например, с помощью рекурсивных процедур на языке АЛГОЛ-60 реализовано послойное покрытие. Этот вариант использовался для решения задач глобальной оптимизации невысокой размерности. Ниже, следуя [4], для покрытия X применим метод ветвей и границ.

С каждым параллелепипедом P_i свяжем набор $S_i = (c_i, d_i, q_i)$. Совокупность наборов S_i для всего множества непокрытых параллелепипедов B_m будем называть списком наборов и обозначать $S = \{S_1, \dots, S_m\}$. Запись $S = \emptyset$ означает, что список S пуст.

Из списка S выбирается тот параллелепипед, с которого начинается процесс покрытия. Это так называемый рабочий параллелепипед. Использовались следующие три правила выбора:

1. Правило MINQ. Берется параллелепипед P_s , такой, что $q_s = \min_{1 \leq i \leq m} q_i$.
2. Правило FIFO. Берется первый параллелепипед из списка, т.е. $s = 1$.
3. Правило LIFO. Берется последний параллелепипед из списка, т.е. $s = m$.

В следующей формулировке алгоритма буква k будет обозначать порядковый номер итерации основного цикла.

Начальные операции. Положить $k = 1, m = 1, P_1 = P$. Задать $\varepsilon > 0$, константу Липшица (либо l , либо L , либо M) и некоторую точку $x_0 \in P$. Вычислить $c_1, w_1, f(c_1), f(x_0), q_1$, определить $\tilde{R} = \min \{f(c_1), f(x_0)\}$. Положить $N_1^{(1)} = \{c_1\}, B_1^{(1)} = \{P_1\}, S_1 = (c_1, w_1, q_1), S^{(1)} = \{S_1\}$. Если $\tilde{R} = f(c_1)$, то в качестве рекордной точки взять c_1 , иначе – точку x_0 . Если $q_1 \geq \tilde{R} - \varepsilon$, то закончить расчеты.

Основной цикл. Повторять следующие шаги 1 – 5, пока $S^{(k)} \neq \emptyset$, т.е. до тех пор, пока множество X не будет полностью покрыто.

1. В соответствии с одним из правил выбора определить рабочий параллелепипед P_s .
2. В параллелепипеде P_s определить t – номер ребра максимальной длины: $w_s = d_s^t$.
3. Разделить параллелепипед P_s пополам по t -й координате, породив тем самым два новых параллелепипеда P' и P'' . Их центры, длины максимальных ребер, минимальные значения минорант обозначим c', w', q' и c'', w'', q'' соответственно. Исключить параллелепипед P_s из набора $B_m^{(k)}$, удалить набор S_s из списка $S^{(k)}$ и положить $m = m - 1$.
4. Вычислить

$$\tilde{R} = \min \{f(c'), f(c'')\}. \quad (32)$$

Если $\tilde{R} \geq R^{(k)}$, то положить $R^{(k+1)} = R^{(k)}$. Если $q' < R^{(k+1)} - \varepsilon$ и $w' \geq \xi$, то включить $S' = (c', w', q')$ в список наборов и положить $m = m + 1$; если

$q'' < R^{(k+1)} - \varepsilon$ и $w'' \geq \xi$, то включить $S'' = (c'', w'', q'')$ в список наборов и положить $m = m + 1$.

Если $\tilde{R} < R^{(k)}$, то положить $R^{(k+1)} = \tilde{R}$. В качестве рекордной точки x_r взять ту из точек c', c'' , в которой достигается минимум (32). Каждый набор S_i из $\{S_i\}_{1 \leq i \leq m}$, удовлетворяющий условию $q_i \geq R^{(k+1)} - \varepsilon$, исключить из этого списка. Новый список наборов $\{S_{i1}, \dots, S_{im}\}$ перенумеровать и обозначить $S^{(k+1)} = \{S_j\}_{1 \leq j \leq m}$.

5. Положить $k = k + 1$.

Заметим, что главная диагональ параллелепипеда, полученного описанным дроблением, не меньше, чем d . С учетом этого факта несложно доказать, что имеет место

Утверждение 1. Пусть в задаче (1) функция f , либо ее первые или вторые производные удовлетворяют условию Липшица на n -мерном прямоугольном параллелепипеде P . Тогда описанный алгоритм за конечное число вычислений значений функции f определяет рекордную точку $x_r \in X_^\varepsilon$.*

После завершения расчетов множество P будет полностью покрыто параллелепипедами P_i , а рекордная точка x_r будет ε -решением задачи (1).

Работу описанного алгоритма существенно ускоряют процедуры локального поиска экстремума. В приведенных ниже результатах расчетов для локального поиска использовался метод сопряженных градиентов с операцией проектирования на параллелепипед P . Программная реализация была осуществлена Н.И. Грачевым.

Приведенный алгоритм можно интерпретировать как метод ветвей и границ [10]:

- на шаге 1 осуществляется выбор множества для ветвления;
- на шаге 2 выбирается способ ветвления;
- на шаге 3 выполняется ветвление;
- на шаге 4 вычисляются оценки q', q'' , отвечающие новым множествам-кандидатам для ветвления, выполняется пересчет рекорда R и производится (если рекорд улучшен) отсев множеств, которые могут быть исключены из области поиска.

На каждом шаге описанного выше метода один параллелепипед из текущего набора разбивается пополам на два новых плоскостью, параллельной координатной. Процесс половинных делений параллелепипедов можно интерпретировать как рост двоичного дерева. Вершинам дерева соответствуют параллелепипеды, полученные при начальном разбиении. Дуги соединяют данный параллелепипед с параллелепипедами, полученными из него в результате деления. Параллелепипеды, отвечающие конечным вершинам дерева, образуют текущий набор параллелепипедов. Некоторые конечные вершины могут быть исключены из этого набора согласно правилу отсева (см. шаг 4).

Для краткости будем далее называть текущий набор параллелепипедов пулом, а действия основного цикла, описываемые шагами 1-5, итерацией.

Эффективность покрытия допустимого множества существенно увеличивается в случае удачного выбора начальной точки, что обеспечивает знание рекорда, достаточно близкого к искомому оптимуму. Для этой цели обычно вначале производятся расчеты с заведомо заниженными константами Липшица. При этом мажоранты и миноранты могут значительно отличаться от функции $f(x)$. Неравенства (20) могут быть при этом не выполнены.

5. Описание параллельного алгоритма

Главная идея параллельного алгоритма состоит в том, чтобы выполнять итерации параллельно на множестве процессоров, с периодическим обменом рекордами и перераспределением областей поиска между ними в процессе счета.

Для параллельного выполнения итераций нужен способ распределения параллелепипедов между процессорами. Предположим, что для расчетов имеется p процессоров. Пусть имеется некоторая начальная последовательность $B_m = \{P_1, \dots, P_m\}$ параллелепипедов P_i , принадлежащих P , например, построенная ранее в процессе работы данного алгоритма. При этом такая начальная последовательность может быть представлена как единственным параллелепипедом ($m = 1$), так и совокупностью s подпоследовательностей параллелепипедов (деревьев). Пусть по (3) вычислены текущий рекорд R_m и текущая рекордная точка x_r .

Способ распределения работы между процессорами свяжем с выбором единицы работы алгоритма, в качестве которой определим вычислительную работу, связанную с Q -кратным ($Q \geq 1$) выполнением итерации последовательного алгоритма.

Пусть помимо p рабочих процессоров имеется еще один процессор, называемый диспетчером. Распределим параллелепипеды начального набора B_m между рабочими процессорами ($p \geq s$).

При выполнении итераций последовательного алгоритма каждый рабочий процессор поддерживает свой индивидуальный пул параллелепипедов, подлежащих исследованию. Будем говорить, что наступает завершение работы процессора, если им выполнены Q циклов последовательного алгоритма (т.е. сделано Q половинных делений) или если его индивидуальный пул стал пустым. Процессор, который завершил работу, сообщает диспетчеру следующие величины: число параллелепипедов в индивидуальном пуле, минимальную оценку y для каждого из них и свой индивидуальный рекорд (т.е. лучшее значение функции f , полученное в ходе вычислений). Этот процессор ставится в поддерживаемую диспетчером очередь ожидающих процессоров. Ожидающий процессор назовем свободным, если его индивидуальный пул пуст.

Рассмотрим метод коммуникации между процессорами. Обмен информацией был бы идеальным, если каждый процессор сообщил бы другим о наилучшем найденном рекорде, о количестве имеющихся параллелепипедов и о значениях нижней оценки для параллелепипедов в текущем индивидуальном наборе. Однако слишком частые обмены между процессорами могут привести к чрезмерному увеличению коммуникационных расходов и понижению эффективности параллельного алгоритма в целом.

В предлагаемом варианте алгоритма каждый процессор посылает диспетчеру вышеперечисленные данные в асинхронном режиме, т.е. независимо от других процессоров. На основе полученных данных диспетчер улучшает рекорд R и

сообщает его всем ожидающим процессорам, которые будут использовать новый рекорд для отсева параллелепипедов из своих наборов. Диспетчер выявляет среди ожидающих процессоров те, которые содержат только один параллелепипед в своем индивидуальном наборе, и им приказывает выполнить основной цикл Q раз, взяв новое значение R в качестве рекорда. Точно такой же приказ получают другие ожидающие процессоры, если в данный момент отсутствуют свободные.

Если имеются еще не распределенные параллелепипеды начального набора, то их подпоследовательности передаются выявленным свободным процессорам.

Если параллелепипеды начального набора все распределены, то как только диспетчер выявит свободный процессор, он среди ожидающих подберет ему партнера для передачи параллелепипеда с минимальной нижней оценкой. Подбор такой пары завершается двумя приказами: свободному процессору дается приказ принять параллелепипед от ожидающего процессора, а ожидающему – передать параллелепипед данному свободному.

Диспетчер завершает выполнение алгоритма, если все рабочие процессоры свободны.

В предлагаемом алгоритме частота обращения процессоров к диспетчеру зависит от параметра Q . Как только найдется свободный процессор, производится перераспределение вычислительной работы между процессорами путем передачи параллелепипедов. При такой схеме организации работы процессоры выполняют алгоритм асинхронно, обмениваясь информацией только при обращении к диспетчеру. В результате процесс выполнения алгоритма становится “недетерминированным”, т.е. последовательность порождаемых параллелепипедов и получаемое решение могут оказаться различными для разных запусков одной и той же задачи (даже при одном и том же наборе процессоров).

Опишем параллельный алгоритм для вышеизложенной схемы взаимодействия процессоров. Для обмена данными между процессорами предполагается использовать явную передачу сообщений. Сообщение, передаваемое от рабочего процессора диспетчеру, содержит информацию о значениях следующих величин:

- число параллелепипедов в текущем пуле;
- минимальную оценку q для параллелепипедов текущего пула;
- индивидуальный рекорд $R = f(c)$.

Диспетчер посылает рабочим процессорам следующие сообщения-команды:

- $work(Q, R)$ – выполнить Q раз основной цикл, считая R текущим рекордом;
- $read(T)$ – принять начальную подпоследовательность параллелепипедов T ;
- $take(i)$ – принять параллелепипед от процессора i ;
- $give(j)$ – передать параллелепипед процессору j ;
- $finish$ – завершить работу.

Алгоритм для процессора-диспетчера:

Шаг 1. Объявить свободными все рабочие процессоры.

Если начальная последовательность задана единственным параллелепипедом ($m = 1$), то объявить процессор p_1 ожидающим и указать, что в его индивидуальном пуле единственный параллелепипед – исходный; принять рекорд $R = f(x_0)$, где $x_0 \in P$ – заданная начальная точка; положить оценку y равной оценке исходного параллелепипеда, задать параметр $Q \geq 1$.

Шаг 2. Если не истекло заданное время работы процессоров, то пока имеются нераспределенные параллелепипеды начального набора и пока все процессоры не стали свободными, повторять следующие шаги 3 – 7.

Шаг 3. Если нет свободных процессоров, то каждому ожидающему дать команду *work*, а список ожидающих опустошить.

Шаг 4. Каждому ожидающему процессору с единственным параллелепипедом в пуле дать команду *work* и удалить этот процессор из списка ожидающих.

Шаг 5. Пока имеются свободные процессоры и нераспределенные параллелепипеды начального набора, выполнять цикл: каждому свободному процессору дать команду *read* и удалить его из списка свободных.

Шаг 6. (Цикл выбора пар процессоров для передачи параллелепипедов.) Пока списки ожидающих и свободных процессоров непусты, выбрать среди ожидающих процессор i с минимальной оценкой, а среди свободных – некоторый процессор j и удалить выбранные процессоры из соответствующих списков; отправить первому команду *give*(j), второму – *take*(i).

Шаг 7. Принять поступающие от процессоров сообщения; занести пославший сообщение процессор в список ожидающих или свободных в зависимости от числа параллелепипедов в его пуле. Пересчитать текущий рекорд с учетом данных, полученных от процессоров.

Шаг 8. Каждому процессору дать команду *finish*.

Алгоритм для рабочего процессора:

Шаг 1. Пока не поступила команда *finish*, выполнять следующие шаги 2 – 4 данного алгоритма.

Шаг 2. Принять очередную команду диспетчера.

Шаг 3. Если получена команда *give* или *take*, то выполнить ее и отправить диспетчеру сообщение о числе своих параллелепипедов и о минимальной оценке y .

Шаг 4. Если получена команда *work*, то выполнять итерации, пока индивидуальный пул непуст и выполнено менее Q итераций; отправить диспетчеру сообщение с результатами работы.

Шаг 5. Если получена команда *read*, то выполнить ее и отправить диспетчеру сообщение о числе параллелепипедов и о минимальной оценке y .

Шаг 6. Если получена команда *finish*, то завершить работу.

6. Результаты вычислительных экспериментов

Для численных экспериментов использовалась функция Морса, определяющая энергию атомных кластеров, состоящих из n атомов:

$$f(x_1, \dots, x_n, \rho) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n ((e^{\rho(1-\|x_i-x_j\|)} - 1)^2 - 1)$$

где ρ - скалярный параметр, $x_i \in R^3$ и $x_j \in R^3$ - трехмерные векторы координат центров атомов i и j соответственно.

Задача состоит в определении декартовых координат n атомов таким образом, чтобы значение функции f достигало глобального минимума. Величина ρ принималась равной 3, 6, 8, 10 и 14, число атомов кластера достигало 85, при этом число искомых скалярных переменных - 255. На фиг. 6 приведена структура кластера для 85 атомов при $\rho = 6$, значение потенциальной энергии составляет -405,246158.

Эксперименты проводились при двух допущениях: функция f удовлетворяет условию Липшица с константой l (стратегия I), ее градиент удовлетворяет условию Липшица с константой L (стратегия II). Так как значения l и L априори неизвестны, то проводился ряд расчетов: вначале с небольшим значением l или L , а после нахождения некоторого рекорда значение l или L постепенно увеличивалось таким образом, чтобы время счета было приемлемым.

Для расчетов использовался вычислительный комплекс МВС-100к Межведомственного суперкомпьютерного центра Российской академии наук [11]. Эксперименты также проводились в распределенной GRID-среде на ресурсах ВЦ РАН и ИСП РАН с использованием технологии Globus 4.0.4. Результаты вычислительных экспериментов для атомных кластеров с числом атомов менее 80 с большой точностью совпали с данными, приведенными в Cambridge Cluster Database (базе данных по атомным кластерам Кембриджа) [12].

Для сравнения эффективности двух стратегий в тестовой функции Морса из 75 переменных на оптимальном значении были зафиксированы все переменные, кроме двух. Вид такой функции приведен на фиг. 1, где изображена зависимость функции Морса f от координат x^{55} и x^{56} со схемой дробления. На фиг. 2 показаны покрывающие параллелепипеды для двух схем дробления (a - по максимальному ребру и b - по масштабируемому ребру) по координатам x^{10} и x^{13} .

Приведенные в таблице 1 результаты расчетов демонстрируют, что стратегия II в несколько раз сокращает число разрезов, необходимое для нахождения оптимума. Кроме того у этой стратегии на порядок выше достоверность оценок, благодаря чему она не отсеивает те параллелепипеды, которые могут содержать оптимум, что повышает вероятность его нахождения.

Таблица 1. Сравнение двух стратегий

Стратегии	Способ выбора ребра	Число параллелепипедов	Процент недостоверных оценок
$f(x_{10}, x_{13})$			
I	a	16 947	14,22%
II	a	7 895	0,00%
II	b	2 097	0,05%
$f(x_4, x_{12})$			
I	a	17 327	1,63%
II	a	7 419	0,05%
II	b	3 781	0,08%
$f(x_0, x_3)$			
I	a	33 761	10,06%
II	a	5 965	1,15%
II	b	4 823	2,26%

На фиг. 3 показан сравнительный график поиска оптимума для функции Морса при 75 свободных переменных и $\rho = 3$ для этих двух стратегий. Из графика видно, что стратегия II значительно быстрее выходит на оптимум.

В таблице 2 приведены результаты сравнения трех способов выбора рабочего параллелепипеда : MINQ (по минимальной оценке), FIFO (в порядке очереди) и LIFO (с “хвоста” очереди).

Таблица 2. Сравнение трех способов выбора параллелепипеда

Характеристики	Правило выбора рабочего параллелепипеда		
	MINQ	FIFO	LIFO
Время решения, сек.	145	49	67
Число параллелепипедов	200 285	78 697	113 967
№ параллелепипеда, где был достигнут оптимум	135 850	30 924	89 606
Максимальный размер списка (число параллелепипедов)	12 821	12 865	22

На фиг. 4 и 5 соответственно приведена зависимость полученного рекорда и числа выполненных разрезов от числа процессоров при фиксированном времени расчетов. Вычислительные эксперименты показали, что с ростом числа процессоров полученный рекорд существенно улучшается, а число разрезов растет прямо пропорционально числу процессоров.

7. Заключение

Описанные в данной статье варианты метода неравномерных покрытий позволяют проводить глобальную оптимизацию липшицевых функций. Авторы сняли условие знания и постоянства константы Липшица на всей области поиска. Введено условие дифференцируемости функции. Благодаря этому, удалось существенно улучшить нижние оценки значений функций в текущих областях поиска и, тем самым, сократить время расчетов, ускорив отсев “неперспективных” параллелепипедов. Предложенный метод параллельного глобального поиска можно применять к решению задач многокритериальной оптимизации. Наиболее простой способ состоит в

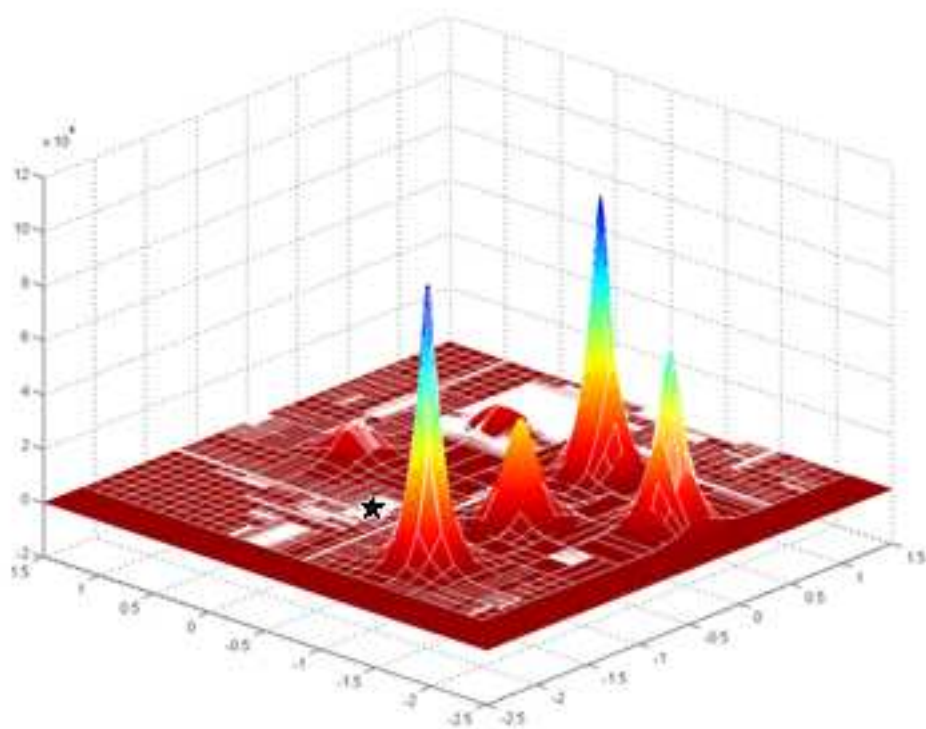
использовании результатов работы [13]. Этот метод также переносится на отыскание решений задачи поиска глобального минимакса аналогично тому, как было сделано в [14]. Кроме того, его можно использовать для минимизации липшицевых функций при дополнительном требовании целочисленности всех или некоторых компонент вектора x .

Авторы выражают благодарность сотруднику ВЦ РАН А.Я. Белянкову за обсуждение статьи и сделанные полезные замечания.

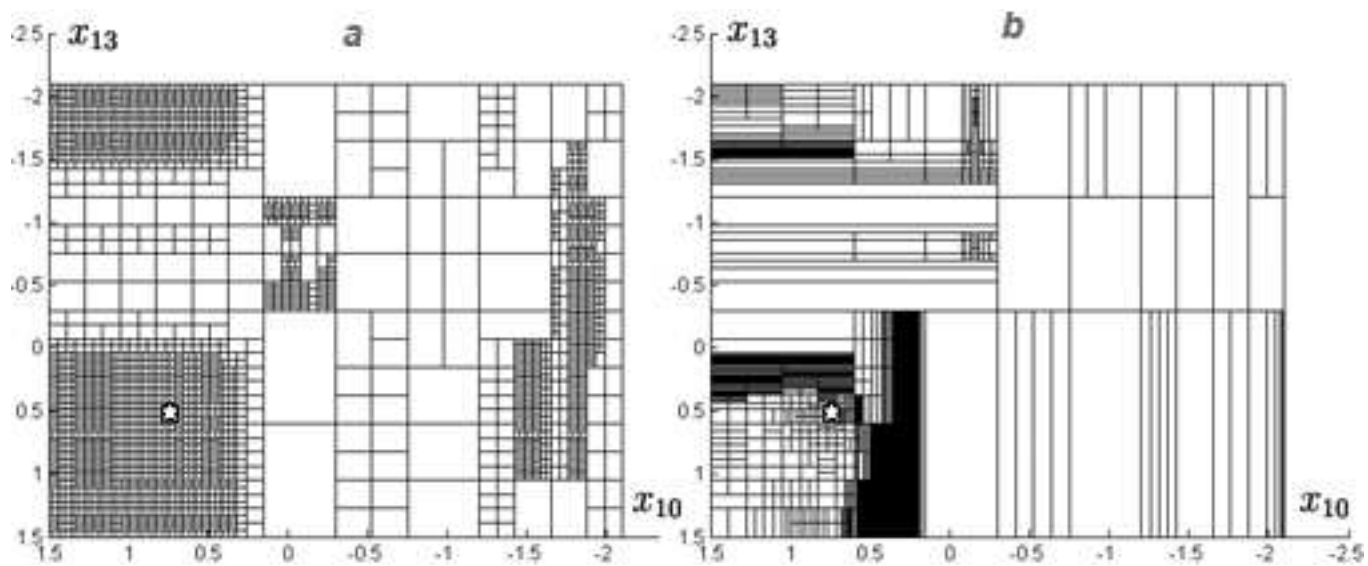
СПИСОК ЛИТЕРАТУРЫ

1. Roman G. Strongin, Yaroslav D. Sergeyev. Global Optimization with Non-Convex Constraints. Sequential and Parallel Algorithms. Dordrecht/Boston/London: Kluwer Academic Publishers, 2000.
2. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002.
3. Евтушенко Ю.Г. Численный метод поиска глобального экстремума функций (перебор на неравномерной сетке) // Ж. вычисл. матем. и матем. физ. 1971. Т. 11. № 6.
4. Евтушенко Ю.Г., Раткин В.А. Метод половинных делений для глобальной оптимизации функции многих переменных // Техн. кибернетика. 1987. № 1. С. 119–127.
5. Белянков А.Я. Повышение вычислительной эффективности методов неравномерных покрытий в глобальной оптимизации. // Методы математического программирования и программное обеспечение. Тез. докл. Свердловск: УрО АН СССР, 1989. С. 21–22.
6. Евтушенко Ю.Г., Малкова В.У., Станевичюс А.А. Распараллеливание процесса поиска глобального экстремума. // Автоматика и телемеханика. 2007. № 5. С. 46–58.
7. Nesterov Yu., Polyak B. Cubic regularization of Newton method and its global performance. // Math. Program. 108(1), 177–205, 2006.
8. Стрекаловский А.С. Элементы невыпуклой оптимизации. Новосибирск: Наука, 2003.
9. Белянков А.Я. Предварительное дробление параллелепипеда в методах неравномерных покрытий в глобальной оптимизации // II Всероссийская научная конференция с молодежной научной школой “Математическое моделирование развивающейся экономики”. ЭКОМОД-2007. Сборник трудов. Киров, изд-во ВятГУ, 2007. С. 59–62.
10. Посыпкин М.А., Сигал И.Х. Исследование алгоритмов параллельных вычислений в задачах дискретной оптимизации ранцевого типа. // Ж. вычисл. матем. и матем. физ. 2005. Т. 45. № 10. С. 1801–1809.
11. Межведомственный суперкомпьютерный центр Российской академии наук. Web-сайт: <http://www.jscc.ru>
12. Cambridge Cluster Database. Web-сайт: <http://www-wales.ch.cam.ac.uk/~jon/structures/Morse.html>
13. Евтушенко Ю.Г., Поталов М.А. Методы решения многокритериальных задач. -ДАН СССР. 1986 Т. 291. № 1. С. 25–29.

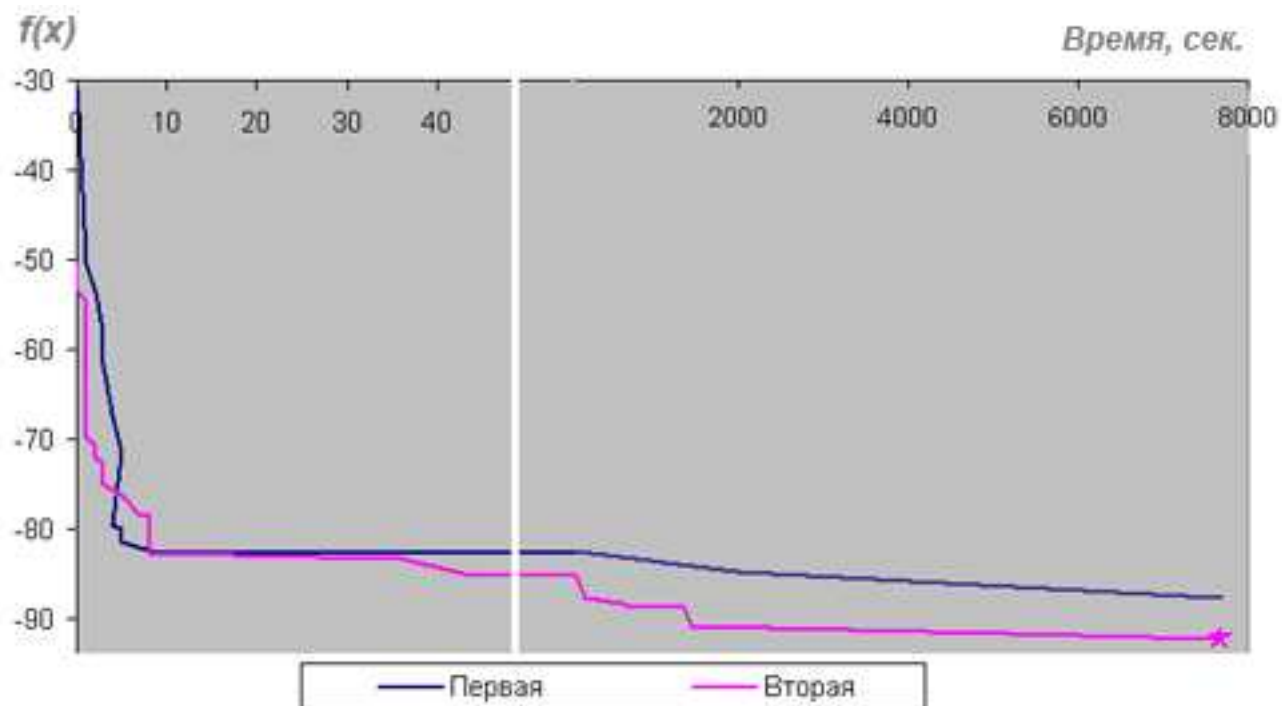
14. *Евтушенко Ю.Г.* Численный метод отыскания наилучших гарантированных оценок. // Ж. вычисл. матем. и матем. физ. 1972 Т. 12. № 1. С. 89–104.



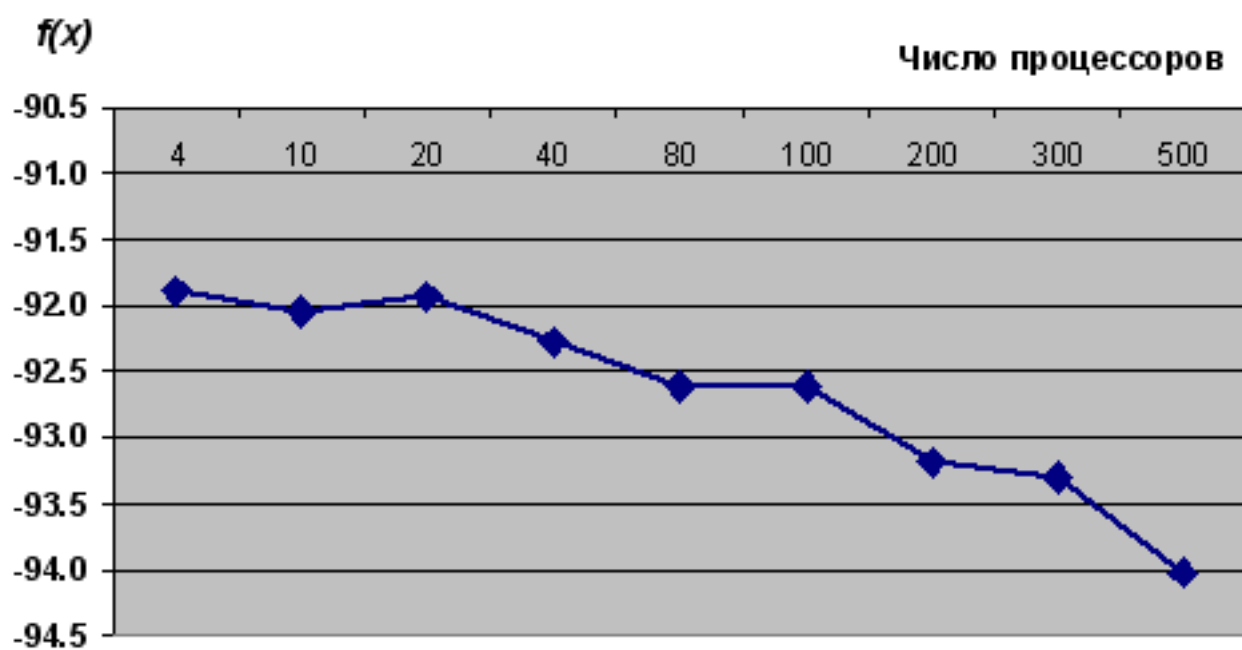
Фиг. 1. Зависимость функции Морса f от координат x^{55} и x^{56} со схемой дробления. Звездочкой указана точка глобального минимума.



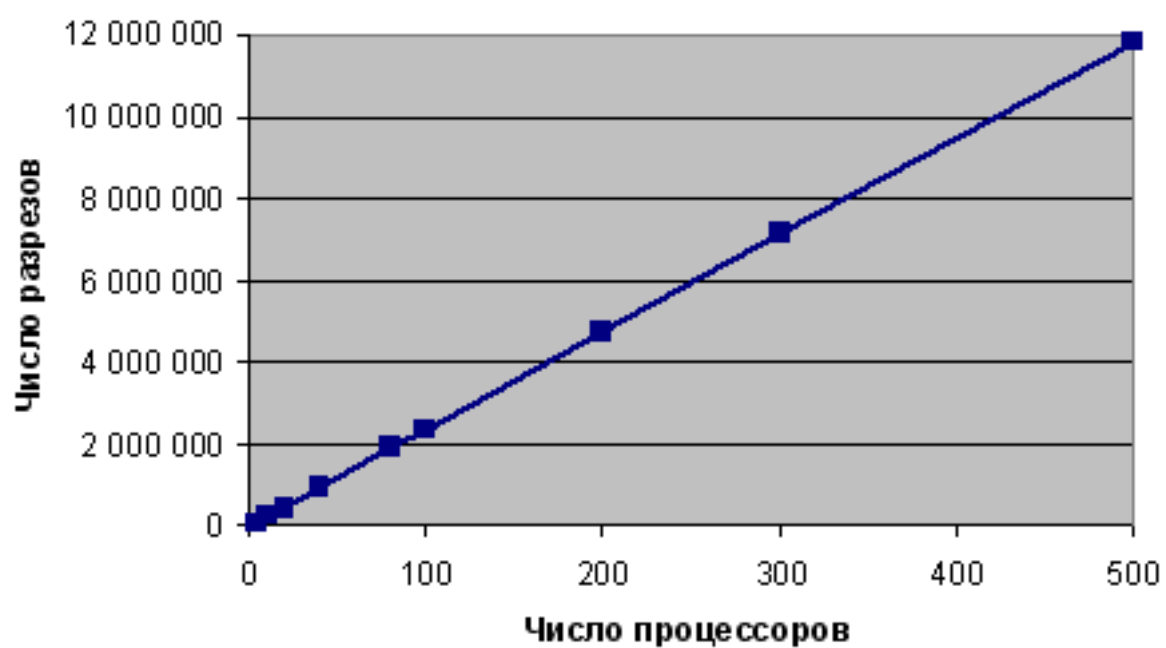
Фиг. 2. Две схемы дробления: по максимальному ребру (a) и по масштабируемому ребру (b). Звездочкой указана точка глобального минимума.



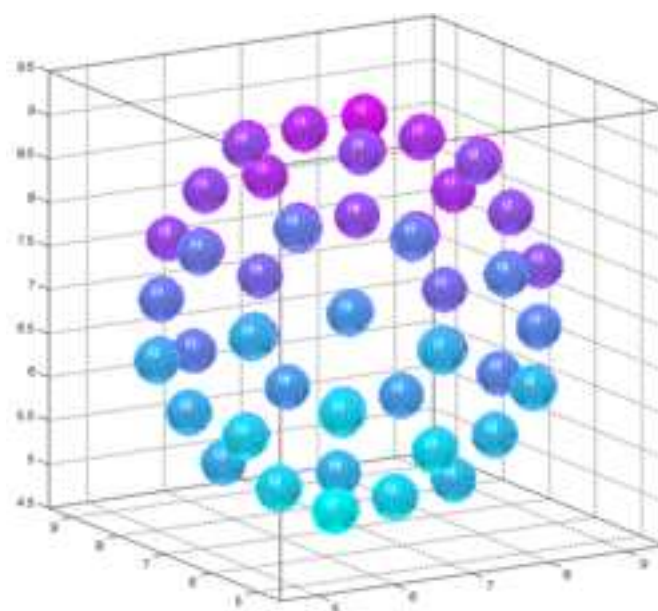
Фиг. 3. Сравнение эффективности первой и второй стратегий. Звездочкой указана точка глобального минимума.



Фиг. 4. Зависимость рекорда $f(x)$ от числа процессоров при фиксированном времени расчета.



Фиг. 5. Зависимость числа разрезов от числа процессоров при фиксированном времени расчета.



Фиг. 6. Структура атомного кластера ($n = 85, \rho = 6$).