

С.К. Завриев, Ю.Н. Перунова

ПАРАЛЛЕЛЬНЫЕ ВЕРСИИ МОДИФИЦИРОВАННЫХ МЕТОДОВ ПОКООРИНАТНОГО И ГРАДИЕНТНОГО СПУСКА И ИХ ПРИМЕНЕНИЕ ДЛЯ РЕШЕНИЯ НЕКОТОРОГО КЛАССА ЗАДАЧ ГЛОБАЛЬНОЙ ОПТИМИЗАЦИИ

1 Введение

Интерес к параллельным алгоритмам глобальной оптимизации вызван необходимостью решать современные задачи повышенной вычислительной сложности в режиме реального времени.

Реализация алгоритмов на многопроцессорных или векторных системах [1]-[4] позволяет проводить независимые расчеты для нескольких траекторий одновременно, а также, в отдельных случаях, сокращать затраты времени, используя дополнительные вычислительные мощности в рамках одной итерации.

Настоящая работа продолжает исследования, начатые в [5],[6] где было рассмотрено поведение траекторий конечно-разностных алгоритмов и выявлены улучшенные свойства глобальной сходимости для класса так называемых γ -регулярных задач глобальной оптимизации.

Одним из методов решения регулярных задач является метод мультистарта [7], [8] с применением стандартных алгоритмов локального поиска, таких как градиентный метод или метод покоординатного спуска. Алгоритмы локального спуска, использующие точные значения градиента функции цели, могут быть не эффективны или не применимы для решения данного класса задач. Величина градиента является точечной характеристикой поведения дифференцируемой функции и чутко реагирует на каждый локальный минимум.

Применение конечных разностей вместо точных значений градиента позволяет сгладить функцию цели и обходить незначительные минимумы на пути к более глубокому из них.

Предлагаемый подход интересен для решения как регулярных, так и нерегулярных задач, функция цели которых обладает большим количеством экстремумов. Описанные алгоритмы способны фильтровать (не замечать) большую часть неглубоких локальных минимумов функции

цели благодаря помехоустойчивым свойствам методов локального спуска 0-го порядка [9] и их параллельных версий [10]. Применение предложенного подхода эффективно для исследования глобального поведения функции цели, но приводит к снижению точности решения задачи в окрестности оптимального множества.

В настоящей статье представлены параллельные версии алгоритма градиентного спуска с использованием конечных разностей и алгоритма покоординатного спуска и исследованы условия их сходимости для класса γ -регулярных задач глобальной оптимизации. Приведены результаты численных экспериментов на основе тестовых функций [11].

Параллельные алгоритмы реализованы на системе Parsytec SuperCluster Gcel-64 ВЦ РАН, г.Москва.

С помощью разработанных алгоритмов проведены расчеты для задачи поиска глобального минимума потенциальной энергии N -атомной молекулы. В качестве целевой функции задачи выбрана функция Леннарда-Джонса [12], $N = 13, 14, 16, 76, 77, 78$.

2 Постановка задачи глобальной оптимизации

Рассмотрим задачу

$$\mathcal{F}(x) \rightarrow \min_{x \in \mathbf{R}_k}, \quad (2.1)$$

где $\mathcal{F} : \mathbf{R}_k \rightarrow \mathbf{R}_1$ многоэкстремальная функция цели, а \mathbf{R}_k -- k -мерное Евклидово пространство.

Требуется найти приближенное решение поставленной задачи, т.е. точку $x \in \mathcal{X}_{opt}(\varepsilon)$ для достаточно малого $\varepsilon \geq 0$, где

$$\mathcal{X}_{opt}(\varepsilon) := \{x \in \mathbf{R}_k | \mathcal{F}(x) \leq f_{opt} + \varepsilon\},$$

$$f_{opt} := \min_{x \in \mathbf{R}_k} \mathcal{F}(x).$$

Функция цели $\mathcal{F}(x)$ в рассматриваемых задачах не обязана быть дифференцируемой и, более того, непрерывной, но ее глобальная структура обладает свойствами регулярной функции. Аналогичный подход применяется в [13],[14].

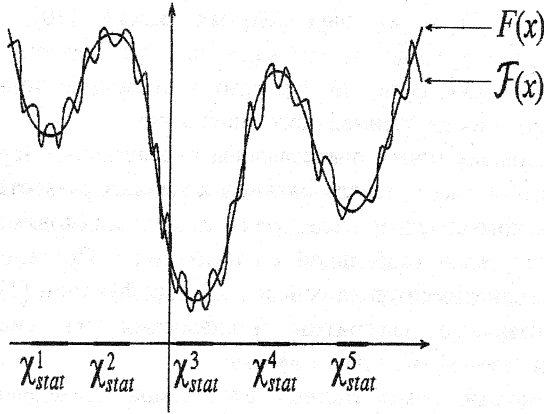


Рис. 1. Функция цели $\mathcal{F}(x)$ γ -регулярной задачи (2.1).

Определение 2.1. Будем называть задачу (2.1) γ -регулярной задачей глобальной оптимизации, если

$$|\mathcal{F}(x) - F(x)| \leq \gamma \quad \forall x \in \mathbf{R}_k, \quad (2.2)$$

где $F(x)$ -- некоторая регулярная функция (см. Рисунок 1), такая, что

1) $F(x)$ непрерывно дифференцируема на \mathbf{R}_k и ее градиент удовлетворяет условию Липшица с константой $L > 0$:

$$\nabla F(\cdot) \in C_{Lip}(\mathbf{R}_k, L);$$

2) множество стационарных точек X_{stat} функции $F(x)$ конечно

$$X_{stat} = \{x_{stat}^1, \dots, x_{stat}^M\}, \quad |X_{stat}| = M;$$

3) в некоторой η -окрестности множества X_{stat} существует гессиан H функции $F(x)$ и он невырожден:

$$\|H^{-1}\| \leq 1/\mu \quad \forall x \in B_\eta(X_{stat}),$$

где

$$B_\varepsilon(X) := \{x \in \mathbf{R}_k \mid \exists x' \in X : \|x - x'\| \leq \varepsilon\};$$

4) множество

$$X_{stat}(\varepsilon) := \{x \in \mathbf{R}_k \mid \|\nabla F(x)\| \leq \varepsilon\},$$

предполагается ограниченным для любого $\epsilon \geq 0$;

5) для некоторого $\epsilon > 0$ и для $\eta > 0$ из п.3) выполнено

$$X_{stat}(\epsilon) \subset B_\eta(X_{stat}). \quad (2.3)$$

Обозначим через \mathcal{X}_{stat}^i окрестность изолированной точки стационарного множества $F(x)$ (см. Рисунок ??):

$$\mathcal{X}_{stat}^i := \{x \in \mathbf{R}_k \mid \|x - x_{stat}^i\| \leq 2 \frac{\sqrt{L}}{\mu} \gamma^{1/2}\}, \quad i = 1, \dots, M. \quad (2.4)$$

Множество $\mathcal{X}_{stat} = \bigcup_{i=1}^M \mathcal{X}_{stat}^i$ будем называть стационарным множеством функции $\mathcal{F}(x)$.

Следующая лемма говорит о том, что близость значений функций $\mathcal{F}(x)$ и $F(x)$ при определенных условиях сохраняет близость стационарных множеств этих функций.

Лемма 2.2 ([6]). Пусть для параметров задачи (2.1), (2.2) $\eta, \epsilon, \mu, L > 0$ и $M \in \mathbf{N} = \{1, 2, \dots\}$ справедливо следующее неравенство:

$$2\sqrt{L}\gamma^{1/2} \leq \epsilon.$$

Тогда

$$\mathcal{X}_{opt}(0) \subset \bigcup_{i=1}^M \mathcal{X}_{stat}^i.$$

На основании Леммы 2.2 можно предположить, что, применяя помехоустойчивые алгоритмы для решения задачи (2.1), при условии (2.2) и небольшом значении γ мы определим приближенно все компоненты множества X_{stat} . Это в свою очередь означает, что при соответствующих значениях параметров η, ϵ, μ, L и M мы получим приближенное решение исходной задачи.

3 Параллельная версия алгоритма покоординатного спуска

Алгоритм покоординатного спуска по сути - последовательный алгоритм, т.к. построение траектории ведется при последовательном переборе всех координат. Поэтому, для создания его параллельной версии необходимо отказаться от последовательного перебора, что в корне изменяет процесс построения траектории. Другими словами, последовательный и

параллельный алгоритмы покоординатного спуска представляют собой два разных алгоритма.

Основная идея представленной параллельной версии алгоритма покоординатного спуска заключается в следующем.

Пусть общее число процессоров равно $(p + 1)$, $p \in \mathbb{N}$. Область определения разделяется на подмножества, принадлежащие различным подпространствам

$$\mathbf{R}_k = \mathbf{R}_{k_1} \times \mathbf{R}_{k_2} \times \dots \times \mathbf{R}_{k_p}.$$

Вычисления на отдельном l -ом процессоре, $l = 1, 2, \dots, p$, ведутся по алгоритму покоординатного спуска в рамках подпространства \mathbf{R}_{k_l} . Такие процессоры будем называть "рабочими". Один дополнительный процессор (с номером $l = p + 1$) координирует действия "рабочих". Этот процессор назовем "управляющим".

Предложенное распределение ролей между процессорами однозначно определяет топологию задачи "звезда" [4].

Алгоритмы $A_1(0)$, $A_1(1)$ являются последовательными. Общее количество процессоров для последовательного алгоритма будем считать равным 1. Для реализации параллельной версии необходимо по крайней мере 3 процессора ($k = 2$, $p = 2$).

Предположим, например, что в наличии имеется неограниченное число процессоров. На практике для проведения вычислений предлагается использовать количество "рабочих" процессоров, равное k - размерности задачи глобальной оптимизации. Каждый такой процессор получит в качестве подпространства одну координату и проверит, есть ли спуск по этой координате. "Управляющий" процессор выберет координату, по которой происходит самый глубокий спуск. В общем виде алгоритм запишется следующим образом.

Алгоритм $A_1(p)$. Параметры: $\theta, 0 < \theta < 1, a > 2\sqrt{\varepsilon_0/L}, \varepsilon_{stop} > 0,$
 $p \in \mathbb{N}, p \leq k, 1 \leq m = \lfloor \frac{k}{p} \rfloor + k - \lfloor \frac{k}{p} \rfloor p.$

Входные данные: $x^0 \in \mathbf{R}_k.$

Результат: $N_{stop} \in \mathbb{N}, x^{N_{stop}} \in \mathbf{R}_k$

Обозначим:

И – инициализация: $n := 0; x^n := x^0.$

П – подготовка к работе: $n := n + 1; x_l^n := x^{n-1} a_l^n := a.$

Выбрать ортонормированную систему $e_i^{m,n}$ из m векторов

$$e_i^{m,n} = \{e_{1,l}^{m,n}, \dots, e_{m,l}^{m,n}\}$$

$$\text{такую, что } e^n = \bigcup_{l=1}^p e_l^{m,n},$$

где

$$e^n = \{e_1^n, \dots, e_k^n\} \in R_k -$$

ортонормированный базис пространства R_k .

P – выполнение на l -ом рабочем процессоре, $l = 1, \dots, p$, одной итерации метода покоординатного спуска (ПС) в рамках подпространства R_{k_l} :

Итерация метода ПС.

Шаг 1. Пусть

$$a_l^n := a/\theta$$

$$\alpha_{l,i}^n := 0, i = 1, \dots, k;$$

$$j_l := 0;$$

Шаг 2. Если $\sum_{i=1}^m |\alpha_{l,i}^n| > 0$, то перейти к *Шагу 5*.

Если

$$a_l^n < \varepsilon_{stop},$$

то перейти к *Шагу 5*, иначе:

$$i := 0$$

$$a_l^n := a_l^n \theta;$$

$$j_l := j_l + 1;$$

Шаг 3. Присвоить $i := i + 1$.

Шаг 3а. Вычислить $\mathcal{F}(x_l^n + a_l^n e_{l,i}^{m,n})$.

Если

$$\mathcal{F}(x_l^n + a_l^n e_{l,i}^{m,n}) > \mathcal{F}(x_l^n) - (a_l^n)^2/2,$$

тогда выполнить *Шаг 3б*,

иначе

$$x_l^n := x_l^n + a_l^n e_{l,i}^{m,n};$$

$$\alpha_{l,i}^n := 1;$$

перейти к *Шагу 4*.

Шаг 3б. Вычислить $\mathcal{F}(x_l^n - a_l^n e_{l,i}^{m,n})$.

Если

$$\mathcal{F}(x_l^n - a_l^n e_{l,i}^{m,n}) > \mathcal{F}(x_l^n) - (a_l^n)^2/2,$$

тогда перейти к *Шагу 4*,

иначе присвоить

$$x_l^n := x_l^n - a_l^n e_{l,i}^{m,n};$$

$$\alpha_{l,i}^n := -1;$$

перейти к *Шагу 4*.

Шаг 4. Если $i = m$, тогда выполнять Шаг 2 ,
иначе Шаг 3.

Шаг 5. Конец итерации.

У – вычисления на $(p + 1)$ -ом управляющем процессоре:

$$l_{n,min} := \arg \min_{l=1,\dots,p} \mathcal{F}(x_l^n);$$

$$x^n := \arg \min_{l=1,\dots,p} \mathcal{F}(x_l^n) = x_{l_{min}}^n;$$

$$a_n = a_{l_{n,min}}^n;$$

$$\alpha_i^n = \alpha_{l_{n,min},i}^n, i = 1, \dots, m;$$

$$\alpha_i^n = 0, i = m + 1, \dots, k.$$

$$j_n = j_{l_{n,min}};$$

К – критерий перехода к следующей итерации: $a_n > \varepsilon_{stop}$;

О – остановка вычислений: $N_{stop} := n$; $x^{N_{stop}} := x^n$.

Блок-схема метода совпадает с предложенной автором в [10] (см.

Рисунок 2).

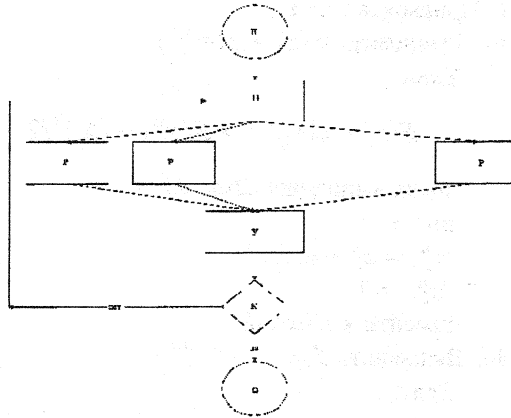


Рис.2. Блок-схема алгоритма $A(p)$.

Очевидно, что для любой траектории данного алгоритма $\{x^n\}$ справедливо

$$\mathcal{F}(x^n) - \mathcal{F}(x^{n+1}) \geq \varepsilon_{stop}^2/2 > 0, n = 0, 1, \dots, N_{stop},$$

поэтому

$$N_{stop} < +\infty.$$

Для некоторой точки $x \in \mathbf{R}_k$ и множества $X \subset \mathbf{R}_k$ обозначим

$$dist(x, X) := \min_{x' \in X} \|x' - x\|.$$

Область сходимости алгоритма $A_1(p)$ для описанного выше класса задач устанавливает следующая теорема.

Теорема 3.1 Пусть параметры алгоритма $A_1(p)$ удовлетворяют неравенствам

$$a > 2\sqrt{\gamma/(L+1)}, \quad \varepsilon_{step} < 2\sqrt{\gamma/(L+1)}, \quad (a)$$

$$\sqrt{k(L+1)}(\sqrt{\theta} + \frac{1}{\sqrt{\theta}})\gamma^{1/2} \leq \epsilon. \quad (b)$$

Тогда для любой траектории $\{x^n\}$ алгоритма $A_1(p)$

$$dist(x^{N_{stop}}, \mathcal{X}_{stat}) \leq r_1 \gamma^{1/2},$$

где

$$r_1 = \frac{\sqrt{L}}{\mu} \left(\sqrt{k(1 + \frac{1}{L})} (\sqrt{\theta} + \frac{1}{\sqrt{\theta}}) - 2 \right).$$

Доказательство. Для доказательства теоремы нам потребуется следующая лемма.

Лемма 3.2 Для любого $x \in \mathbf{R}_k$, $\forall i = 1, \dots, k$ и a , $0 < a \leq \frac{1}{L}$, справедливо

$$| \langle \nabla F(x), a e_i \rangle | \leq \frac{La^2}{2} + F(x) - F(x - \text{sign}(\nabla F(x))_i a e_i),$$

где $\text{sign}(\nabla F(x))_i$ – знак i -ой компоненты вектора $\nabla F(\cdot)$.

Доказательство леммы. Искомое неравенство получаем в подстановке $u = x - \text{sign}(\nabla F(x))_i a e_i$, $v = x$ из леммы [18], стр.100.

Доказательство теоремы. Фиксируем произвольную траекторию $\{x^n\}$ алгоритма $A_1(p)$.

Очевидно, что для любого $i = 1, \dots, k$

$$\mathcal{F}(x^{N_{stop}} \pm a\theta^j e_i) \geq \mathcal{F}(x^{N_{stop}}) + \frac{(a\theta^j)^2}{2}, \quad j = 0, 1, \dots, \tilde{j}_*,$$

где

$$\tilde{j}_* = \min\{j \in \mathbf{N} \mid \theta^j a < \varepsilon_{stop}\}.$$

Следовательно,

$$F(x^{N_{stop}} \pm a\theta^j) \geq F(x^{N_{stop}}) - 2\gamma - \frac{(a\theta^j)^2}{2}, \quad j = 0, 1, \dots, \tilde{j}_*.$$

Поэтому, используя Лемму 3.2, для любого $i = 1, \dots, k$ получаем

$$|\langle \nabla F(x^{N_{stop}}), e_i \rangle| \leq \frac{2\gamma}{a\theta^j} + \frac{(L+1)a\theta^j}{2}.$$

Тогда

$$\|\nabla F(x^{N_{stop}})\| \leq \min_{j=0,1,\dots,\tilde{j}_*} \sqrt{k} \left(\frac{2\gamma}{\theta^j a} + \frac{L+1}{2} \theta^j a \right).$$

При условии (а) из Леммы 4.8 [5] следует

$$\min_{j=0,1,\dots,\tilde{j}_*} \left(\frac{2\gamma}{\theta^j a} + \frac{(L+1)}{2} \theta^j a \right) \leq \sqrt{\gamma(L+1)} \left(\sqrt{\theta} + \frac{1}{\sqrt{\theta}} \right),$$

$$j_* := \min\{j \in \mathbf{N} \mid \theta^j a < 2\sqrt{\gamma/(L+1)}\},$$

Так как $\tilde{j}_* \geq j_*$, то

$$\|\nabla F(x^{N_{stop}})\| \leq \sqrt{k(L+1)} \left(\sqrt{\theta} + \frac{1}{\sqrt{\theta}} \right) \gamma^{1/2}.$$

Применяя условие (б) к (2.3), получим, что существует $i, 1 \leq i \leq M$, такой, что

$$x^{N_{stop}} \in B_\eta(x_{stat}^i).$$

То есть

$$\|x^{N_{stop}} - x_{stat}^i\| \leq \frac{\sqrt{k(L+1)}}{\mu} \left(\sqrt{\theta} + \frac{1}{\sqrt{\theta}} \right) \gamma^{1/2}.$$

Требуемый результат следует из определения \mathcal{X}_{stat} в (2.4).

Теорема доказана.

4 Параллельная версия модифицированного алгоритма градиентного спуска

В алгоритме градиентного спуска вместо вычисления точного градиента функции \mathcal{F} (по условиям задачи (2.1), (2.2) функция цели может быть не дифференцируема) предлагается подобрать приближенное значение градиента с использованием конечных разностей и шага дискретизации пространства $h > 0$.

Вычисления вектора приближенного градиента $\mathcal{G}(x, h)$ в точке x для различных значений параметра h , $0 < h_1 < h < h_p$ возможно выполнить независимо друг от друга. Таким образом, параллельные вычисления по модифицированному алгоритму градиентного спуска позволяют аппроксимировать направление поиска решения задачи оптимизации, затратив времени не больше, чем на вычисление точного градиента в последовательном алгоритме градиентного спуска при решении гладких задач оптимизации.

Аналогично параллельному алгоритму покоординатного спуска для распределения вычислений на несколько процессоров в данном случае используется топология звезды (см. Рисунок 2). Общее число процессоров равно $(p + 1)$.

Для вычисления одной итерации алгоритма на каждом l -ом рабочем процессоре $l = 1, \dots, p$ в качестве возможного направления спуска функции \mathcal{F} в точке x используется вектор $-\mathcal{G}(x, h_l)$.

$$\mathcal{G}(x, h_l) := (\mathcal{G}(x, h_l)_1, \dots, \mathcal{G}(x, h_l)_k), \quad (4.1)$$

$$\mathcal{G}(x, h_l)_i := \{\mathcal{F}(x + h_l e_i) - \mathcal{F}(x)\} / h_l, \quad i = 1, \dots, k,$$

где $\{e_i \mid i = 1, \dots, k\}$ -- ортонормированный базис \mathbf{R}_k , а $h_l > 0$ -- различные значения параметра дискретизации пространства h , $l = 1, \dots, p$.

Выбор наиболее подходящего параметра $h_l > 0$ осуществляет управляющий $(p + 1)$ -ый процессор по результатам спуска в рамках одной итерации.

Алгоритм $A_2(p)$.

Параметры: $h_0, \alpha_0 > 0, S_0, N_0 \in \mathbf{N}, 0 < \theta < 1, \varepsilon_{stop} > 0,$

$$\alpha_0 < \frac{1}{2L}, \quad N_0 \alpha_0 > \frac{1}{L}, \quad p \in \mathbf{N}.$$

Входные данные : $x^0 \in \mathbf{R}_k$

Результат: $N_{stop} \in \mathbf{N}$, $x^{N_{stop}} \in \mathbf{R}_k$

Обозначим:

И – инициализация: $n := 0$; $x^n := x^0$.

П – подготовка к работе: $n := n + 1$; $x^n := x^{n-1}$.

Р – вычисления на l -ом рабочем процессоре, $l = 1, \dots, p$:

$$(a_l, h_l) := \arg \min_{\substack{a=i\alpha_0, i=0, \dots, N_0; \\ h=h_0\theta^j, j=(l-1)S_0, \dots, lS_0-1}} \mathcal{F}(x^n - a\mathcal{G}(\mathcal{F}, x^n, h));$$

У – вычисления на $(p + 1)$ -ом управляющем процессоре:

$$(a^n, h^n) := \arg \min_{l=1, \dots, p} \mathcal{F}(x^n - a_l \mathcal{G}(x^n, h_l));$$

$$x^n := x^n - a^n \mathcal{G}(x^n, h^n)$$

К – критерий останковки:

$$\mathcal{F}(x^n) - \mathcal{F}(x^{n+1}) < \varepsilon_{stop},$$

О – остановка вычислений: $N_{stop} := n$; $x^{N_{stop}} := x^n$.

Блок-схема метода изображена на Рисунке 2.

Нетрудно заметить, что для любой траектории $\{x^n\}$ алгоритма $A_2(p)$ мы имеем

$$\mathcal{F}(x^n) - \mathcal{F}(x^{n+1}) \geq \varepsilon_{stop} > 0, \quad n = 0, 1, \dots, N_{stop} - 1.$$

Так как $\mathcal{F}(x)$ – функция γ -регулярной структуры, то

$$\min_{x \in \mathbf{R}_k} \mathcal{F}(x) > -\infty,$$

следовательно

$$N_{stop} < +\infty.$$

Область сходимости алгоритма $A_2(p)$ для описанного выше класса задач устанавливает следующая теорема.

Теорема 4.1 Пусть параметры алгоритма $A_2(p)$ удовлетворяют условиям

$$h_0 > 2\sqrt{\gamma/L}, \quad (S_0 p - 1) \geq \ln(2\sqrt{\gamma/L}/h_0)/\ln\theta, \quad (a)$$

кроме того

$$kL(\sqrt{\theta} + \frac{1}{\sqrt{\theta}})^2 \gamma + 4LC_0 \gamma + 4L\varepsilon_{stop} \leq \epsilon^2, \quad (б)$$

где $C_0 := 1/(1 - 2L\alpha_0)$.

Тогда для любой траектории $\{x^n\}$ алгоритма ${}_2(p)$ справедливо

$$\text{dist}(x^{N_{\text{stop}}}, \mathcal{X}_{\text{stat}}) \leq r_2 \gamma^{1/2} + \frac{2\sqrt{L}}{\mu} \varepsilon_{\text{stop}}^{1/2},$$

где

$$r_2 = \frac{\sqrt{L}}{\mu} \left((4C_0 + k(\sqrt{\theta} + \frac{1}{\sqrt{\theta}})^2)^{1/2} - 2 \right).$$

Доказательство Для доказательства теоремы нам потребуется следующая лемма.

Лемма 4.2 Для любых $x, \delta \in \mathbf{R}_k$ и $a, 0 < a \leq 1/L$, справедливы оценки:

$$\begin{aligned} & F(x) - F(x - a(\nabla F(x) + \delta)) \geq \\ & \geq a(\|\nabla F(x)\| + \|\delta\|)((1 - aL/2)\|\nabla F(x)\| - (aL/2)\|\delta\|). \end{aligned}$$

Доказательство леммы. Искомое неравенство получаем в подстановке $u = x - a(\nabla F(x) + \delta)$, $v = x$ из леммы [18], стр.100.

Фиксируем произвольную траекторию $\{x^n\}$ алгоритма ${}_2(p)$.

Из определения алгоритма ${}_2(p)$ имеем

$$\max_{\substack{\alpha=i\alpha_0, i=0, \dots, N_0, \\ h=h_0\theta^j, j=0, \dots, S_0p-1}} (\mathcal{F}(x^{N_{\text{stop}}}) - \mathcal{F}(x^{N_{\text{stop}}} + \alpha \mathcal{G}(x^{N_{\text{stop}}}, h))) \leq \varepsilon_{\text{stop}}.$$

Тогда для любого $h = h_0\theta^j, j = 0, \dots, S_0p - 1$

$$F(x^{N_{\text{stop}}}) - \min_{\alpha=i\alpha_0, i=0, \dots, N_0} F(x^{N_{\text{stop}}} - \alpha(\nabla F(x^{N_{\text{stop}}}) + \delta_{\text{stop}}(h))) \leq \varepsilon_{\text{stop}} + 2\gamma,$$

где

$$\delta_{\text{stop}}(h) = -(\nabla F(x^{N_{\text{stop}}}) - \mathcal{G}(x^{N_{\text{stop}}}, h)).$$

Заметим, что

$$\|\delta_{\text{stop}}(h)\| \leq \sqrt{k}(Lh/2 + 2\gamma/h).$$

Тогда при условии а) из Леммы 4.8 [5] следует

$$\min_{\substack{h=\theta^j h_0, \\ j=0, 1, \dots, S_0}} \|\delta_{\text{stop}}(h)\|^2 \leq k\gamma L(\sqrt{\theta} + \frac{1}{\sqrt{\theta}})^2 \quad \forall x \in \mathbf{R}_k.$$

С другой стороны из *леммы 3.2* получаем, что для любых $x, \delta \in \mathbf{R}_k$ и α_0 , $0 < \alpha_0 < 1/(2L)$, $N_0, \alpha_0 N_0 < 1/L$ выполняется:

$$\begin{aligned} F(x) - \min_{0 \leq i \leq N_0} F(x - i\alpha_0(\nabla F(x) + \delta)) &\geq \\ &\geq (1/(2LC_0))(\|\nabla F(x)\|^2 - \|\delta\|^2). \end{aligned}$$

Следовательно

$$\begin{aligned} 1/(2LC_0)(\|\nabla F(x^{N_{stop}})\|^2 - \|\delta_{stop}(h)\|^2) &\leq \varepsilon_{stop} + 2\gamma \\ \forall h = h_0\theta^j, j = 0, \dots, S_0p - 1, \end{aligned}$$

или

$$\|\nabla F(x^{N_{stop}})\|^2 \leq 2LC_0\varepsilon_{stop} + 4LC_0\gamma + kL(\sqrt{\theta} + \frac{1}{\sqrt{\theta}})^2\gamma.$$

Теперь применяем условие (б) и п.4 определения 2.1, откуда следует существование такого i , $1 \leq i \leq M$, что

$$x^{N_{stop}} \in B_\eta(x_{stat}^i).$$

Далее

$$\|x^{N_{stop}} - x_{stat}^i\| \leq \frac{\sqrt{L}}{\mu}(4C_0 + k(\sqrt{\theta} + \frac{1}{\sqrt{\theta}})^2)^{1/2}\gamma^{1/2} + \frac{2\sqrt{L}}{\mu}\varepsilon_{stop}^{1/2}.$$

И требуемый результат следует из определения множества \mathcal{X}_{stat} в (2.4).

Теорема доказана.

5 Численный эксперимент.

При выборе метода решения задачи оптимизации основным вопросом является его функциональная эффективность. Другими словами, преимуществом обладает метод, предлагающий лучшее решение. И следующим этапом является поиск метода, решающего поставленную задачу оптимизации в кратчайший срок. Поэтому, при исследовании эффективности параллельных версий фильтрующих алгоритмов большее внимание было уделено точности решения задачи, а не измерению временных затрат.

Численное сравнение алгоритмов $A_1(p)$, $A_2(p)$, $p \geq 1$ является наиболее ярким показателем эффективности алгоритмов, потому, что отражает не столько скорость пересылки данных между процессорами, а свидетельствует о влиянии правильного подбора параметра дискретизации h^n на погрешность при решении задачи оптимизации.

Алгоритмы серии $A_1(p)$ и $A_2(p)$ были реализованы средствами языков программирования Fortran и С и использованы для решения ряда тестовых задач оптимизации [11], а также задачи структурной химии. Ниже представлены результаты тестирования и сравнения эффективности последовательных и параллельных версий алгоритмов.

Введем следующие обозначения:

$\{x^n\}_1$, $n = 0, 1, \dots, N_{stop}$ - траектория алгоритма покоординатного спуска $A_1(1)$.

$\{x^n\}_2$, $n = 0, 1, \dots, N_{stop}$ - траектория алгоритма наискорейшего спуска A_2 .

\tilde{x}_1^{n+1} - результат выполнения одной итерации алгоритма $A_1(p)$ из точки x_1^n траектории $\{x^n\}_1$.

\tilde{x}_2^{n+1} - результат выполнения одной итерации алгоритма $A_2(p)$ из точки x_2^n траектории $\{x^n\}_2$.

$$d_i(n)_1 = \frac{\mathcal{F}_i(x_1^n) - \mathcal{F}_i(\tilde{x}_1^{n+1})}{\mathcal{F}_i(x_1^n) - \mathcal{F}_i(\tilde{x}_1^{n+1})}, \quad i = 1, \dots, 4;$$

$$d_i(n)_2 = \frac{\mathcal{F}_i(x_2^n) - \mathcal{F}_i(\tilde{x}_2^{n+1})}{\mathcal{F}_i(x_2^n) - \mathcal{F}_i(\tilde{x}_2^{n+1})}, \quad i = 1, \dots, 4, \text{ где } \mathcal{F}_i - \text{ тестовые функции из [11].}$$

$$\mathcal{F}_1(x) = x_1^2 + x_2^2, \quad x^0 = (234, 567);$$

$$\mathcal{F}_2(x) = \frac{1}{50} \sum_{i=1}^2 (x_i - 4.5)(x_i - 3.0)(x_i + 2.0)(x_i + 3.0) *$$

$$*(x_i + 1.5)(x_i - 1.0)(x_i + 4.5)(x_i - 3.7); \quad x_0 = (234, 567);$$

$$\mathcal{F}_3(x) = \frac{x_1^6}{6} - \frac{3}{4}x_1^4 + 2x_1^2 + \frac{1}{2}x_2^2 - x_2x_1^2 + 10; \quad x_0 = (234, 567);$$

$$\mathcal{F}_4(x) = \frac{1}{2} \sum_{i=1}^{10} (x_i^4 - 16x_i^2 + 5x_i);$$

$$x_0 = (10, 20, 30, 40, 50, 60, 70, 80, 90, 100).$$

На Рисунке 3 графически изображено изменение $d_i(n)_1$ в зависимости от номера итерации.

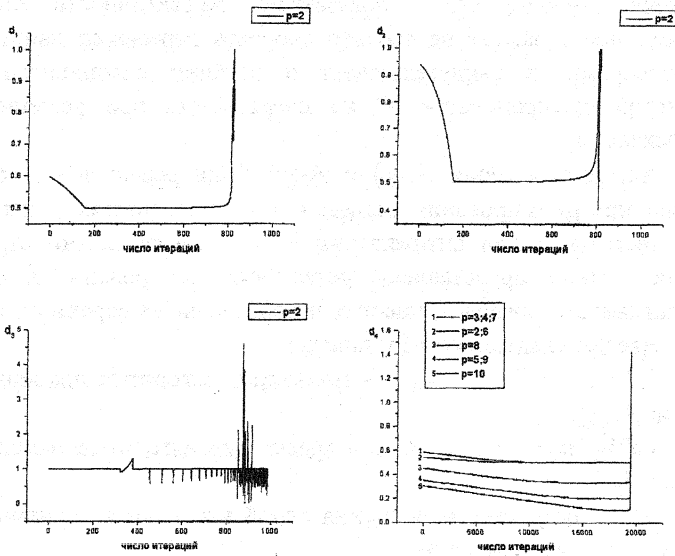


Рис. 3. Расчет тестовых задач на p процессорах. Сравнение результатов работы алгоритмов $A_1(p)$ и A_1 в рамках одной итерации.

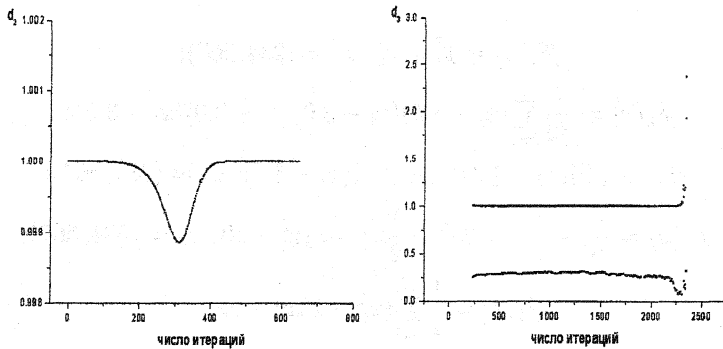


Рис. 4. Расчет тестовых задач при $p = 64$. Сравнение алгоритмов $A_2(p)$ и A_2 в рамках одной итерации.

Можно видеть, что $d_i(n)_1 < 1$, т.е. работа одной итерации алгоритма $A_1(p)$, $p > 1$ приносит лучший результат, чем итерация алгоритма $A_1(1)$.

В близкой окрестности множества решений задачи эффективность параллельной версии алгоритма покоординатного спуска ухудшается. Это связано с тем, что данный алгоритм требует много времени для дробления шага a_n .

Стоит отметить, что большинство всплесков $d_i(n)_1 > 1$ неустойчивы и сопровождаются более глубокими провалами. Так величина самого высокого пика $d_3(880)_1 < 4.6$, а следующего за ним самого низкого провала $d_3(891)_1 < \frac{1}{5.2}$.

На Рисунке 4 графически изображено изменение $d_2(n)_2$ и $d_3(n)_2$ в зависимости от номера итерации. Графики $d_1(n)_2$ и $d_4(n)_2$, соответствующие тестовым задачам под номерами 1 и 4, представляют собой константы, равные 1.

6 Задача Леннарда-Джонса.

Рассмотрим задачу поиска минимума функции потенциальной энергии скопления N молекул.

$$V_{LJ}(N)(r_1, \dots, r_N) \rightarrow \min_{\substack{r_i \in \mathbf{R}_3, \\ i=1, \dots, N}}$$

где

$$V_{LJ}(N)(r_1, \dots, r_N) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N v_{LJ}(\|r_i - r_j\|).$$

Энергия попарного взаимодействия молекул описывается следующей формулой

$$v_{LJ}(r) = r^{-12} - r^{-6}, r \in \mathbf{R}_1,$$

где \mathbf{R}_k - k -мерное евклидово пространство, $\|\cdot\|$ - норма в \mathbf{R}_k . Впервые эта задача была исследована в 1987 году [12]. С того времени изучению данной проблемы посвящено большое количество работ. Их обзор можно найти в [15]. В настоящее время решение выбранной нами и подобных ей задач представляет широкий интерес [16], [17].

Отметим некоторые особенности $V_{LJ}(N)$.

N	p	лучший известный результат $\mathcal{F}_{N_{stop}}$	лучший результат выполнения среди $A_1(p)$ и $A_2(p)$ $\mathcal{F}_{N_{stop}}$
13	16	-44.327	-44.327
14	16	-47.845	-47.845
16	32	-56.816	-56.816
55	64	-279.248	-263.438
76	64	-402.385	-399.151

Табл. 1. Результаты численного решения задачи Леннарда-Джонса методом мультистарта с помощью алгоритмов $A_1(p)$ и $A_2(p)$.

1) С точки зрения глобальной оптимизации функция $V_{LJ}(N)$ чрезвычайно сложна. Количество локальных минимумов по оценкам [19] равно $O(e^{N^2})$.

2) Функция $V_{LJ}(N)$ непрерывно дифференцируема бесконечное количество раз при $r_i \neq r_j, i, j = 1, \dots, N, j \neq i$.

3) Функцию $V_{LJ}(N)$ можно представить в виде

$$V_{LJ}(N)(a) = V_{LJ}(N)(ar_1, \dots, ar_N) = C_1 a^{-12} - 2C_2 a^{-6},$$

где

$$C_1 = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \|r_i - r_j\|^{-12},$$

$$C_2 = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \|r_i - r_j\|^{-6}.$$

Минимум этой, выпуклой по параметру a , функции достигается при $a_{min} = (C_2/C_1)^{1/6}$,

$$V_{LJ}(N)(a_{min}) = C_1((C_1/C_2)^2 - 2) \quad (6.1)$$

Стоит отметить большую неустойчивость вычисления $V_{LJ}(N)(a_{min})$ по формуле (6.1) к ошибкам округления при операциях с плавающей точкой, что приводит к необходимости заменить аналитический способ вычисления a_{min} по (6.1) на его численный эквивалент.

Начальные приближения для расчетов были подготовлены на основе значений аргумента при $(N - 1)$, вычисляемых по алгоритму из [15], стр. 435.

В Таблице 1 представлены численные результаты решения задачи Леннарда-Джонса.

7 Заключение.

Опишем схематично метод решения задачи (2.1) при условии (2.2).

I. Используя в градиентном методе или методе покоординатного спуска величины, относящиеся к $\mathcal{F}(x)$, вместо точных значений функции и градиента $F(x)$, возможно достичь окрестность стационарных точек задачи глобальной оптимизации функции цели $F(x)$.

II. При соответствующих параметрах регулярности η , ϵ , μ , L , γ и M на основании леммы 2.2 можно утверждать, что полученное приближенное решение задачи глобальной оптимизации функции цели $F(x)$ является приближенным решением задачи (2.1).

Заметим, что $\mathcal{F}(x)$ не обязательно дифференцируема и вычисление ее точного градиента становится бессмысленным. Причем соответствующая $\mathcal{F}(x)$ регулярная скелетная функция $F(x)$ явно не определяется в условии задачи и может оставаться неизвестной в ходе всего решения.

Параллельные версии алгоритмов покоординатного спуска разностной аппроксимации градиента по сравнению со своими последовательными аналогами обладают большей вычислительной мощностью и в p раз тщательнее исследует область текущего значения аргумента x^n .

Это преимущество параллельных версий особенно ярко проявляется при сравнении в рамках одной итерации.

При реализации алгоритмов $A_1(p)$, $A_2(p)$, $p > 1$ в реальном времени возникают дополнительные затраты на пересылку данных между процессорами. Величина таких затрат на синхронизацию зависит от количества итераций и скоростных характеристик выбранного вычислительного комплекса.

Приверженность параллельного метода покоординатного спуска большим шагам вдали от окрестности решения является положительным качеством метода, применяемого для решения многоэкстремальных, сильно зашумленных задач.

Литература

1. J.E.Dennis,Jr, V.Torczon *Direct search methods on parallel machines*. SIAM J.Optimization, 1, 4, 448-474. 1991.
2. G.Xue *Molecular Conformation on the CM-5 by Parallel Two-Level Simulated Annealing*. Journal of Global Optimization, 1994, 4, pp. 187-208.
3. Torn A. and Viitanen S. *Topographical global optimization using pre-sampled points*. Journal of Global Optimization, 1994, 5 pp. 267-276.
4. Bertsekas D., Tsitsiklis *Parallel and Distributed Computation: Numerical methods*. Prentice Hall, Englewood Cliffs, N.J.
5. Zavriev S.K. *On the global optimization properties of finite-difference local descent algorithms*. Journal of Global Optimization, 1993, 3, pp 63-78.
6. Завриев С.К., Перунова Ю.Н. *Об использовании конечных разностей в методе мултистарта для решения некоторого класса задач глобальной оптимизации*. Вестник Московского Университета, серия 15, N 3, 1997.
7. Rinnooy Kan A.H., Timmer G.T. *Stochastic global optimization methods*. Mathematical programming. 1987, 39, pp. 27-78.
8. Torn, A., Zilinskas, A. *Global Optimization*. Springer Verlag, Berlin, 1989.
9. Поляк, Б.Т. *Введение в оптимизацию*. М.: Наука, 1983.
10. Перунова Ю.Н. *Помехоустойчивость методов разностной аппроксимации градиента при распараллеливании*. электронный журнал Исследовано в России, 156, с.1782-1791, 2001, <http://zhurnal.ape.relarn.ru/articles/2001/156.pdf>.
11. Yong Y. *Dynamic Tunneling Algorithm for Global Optimization*. IEEE Transactions on systems, man, and cybernatics, 19,5, pp. 1222-1230, 1989.

12. Northby J.A. *Structure and binding of Lennard-Jones clusters: $13 \leq N \leq 147$* . Journal of Chemical Physics 87(10), 15 November 1987, pp 6166-6177.
13. Hu, T. C., Klee, V., and Larman, D. *Optimization of Globally Convex Functions* SIAM J. Control and Optimization 27 (5), 1026-1047, 1989.
14. Mayne, D. Q. and Polak, E. *Nondifferential Optimization via Adaptive Smoothing*. JOTA 43 (4), 601-614, 1984.
15. Maranas C.D. and Floudas C.A. *A deterministic global approach for molecular structure determination*. Journal of Chemical Physics 100(2), 15 January 1994, pp 1247-1261.
16. R.Byrd, E.Eskow, R.Schnabel *A New Large-Scale Global Optimization Method and its Application to Lennard-Jones Problems*. Technical Report CU-CS-630-92, Department of Computer Science, University of Colorado at Boulder.
17. Neumaier A. *Molecular Modeling of Proteins and Mathematical Prediction of Protein structure*. SIAM Review, 39 3, 407-460, 1997.
18. Васильев Ф.П. *Численные методы решения экстремальных задач*. Москва, Наука, 1980.
19. Hoare M. *Structure and Dynamics of Simple Microclusters*. Advances in Chemical Physics 40, pp. 49-135, 1979.