

Применение проективного покоординатного спуска в задаче Фекете

Поляк Б. Т.
ИПУ РАН, Москва

Фатхуллин И. Ф.
МФТИ, Москва

21 апреля 2019 г.

Аннотация

Рассматривается задача минимизации энергии системы из N точек, на поверхности сферы в \mathbb{R}^3 , и взаимодействующих с потенциалом $U = \frac{1}{rs}$, $s > 0$, где r - евклидово расстояние между парой точек. В работе предлагается метод проективного покоординатного спуска, использующий быстрый счет функции и градиента, а также покоординатный метод второго порядка, который достаточно быстро приближается к известным из литературы минимальным значениям.

Ключевые слова: минимизация энергии на сфере, задача Фекете, задача Томсона, проективный покоординатный спуск

1 Введение

Пусть задано натуральное $N \geq 2$. Задача Фекете порядка N формулируется следующим образом [1]. Найти множество точек $\mathbf{x} = \{x_1, \dots, x_N\} \in \mathbb{R}^{3N}$ с ограничением для каждой точки x_i на единичной сфере $S_i^2 = S^2 = \{x \in \mathbb{R}^3 : \|x\|_2 = 1\}$, такое, что оно минимизирует потенциальную энергию системы:

$$U(\mathbf{x}) := \sum_{1 \leq i < j \leq N} U(x_i, x_j) \quad (1)$$

То есть формально решается задача:

$$\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x} \in \prod_{i=1}^N S_i^2} U(\mathbf{x}), \quad (2)$$

где $U(x_i, x_j)$ - функция взаимного потенциала между точками x_i и x_j , которую обычно можно представить в виде ядерной функции от евклидова расстояния $r_{ij} = \|x_i - x_j\|_2$: $\tilde{U}(r_{ij}) = U(x_i, x_j)$. Особый интерес представляет изучение ядерной функции Рисса $\tilde{U}_s(r_{ij}) = r_{ij}^{-s}$, $s > 0$ и логарифмического ядра $\tilde{U}_0(r_{ij}) = -\log(r_{ij})$, которое получается при $s \rightarrow 0$. Далее в работе под $U(\mathbf{x})$ будет подразумеваться функция (1) с ядром $\tilde{U}_s(r_{ij})$, $s \geq 0$.

При $s = 1$ получается кулоновский потенциал и соответственно задача Томсона. Последняя была поставлена Дж. Дж. Томсоном еще в 1904 году [2], однако до сих пор остается нерешенной для произвольного N . Задача Фекете привлекла особое внимание математиков, потому что она появляется в списке Смейла [3] - нерешенных задач для 21-го века. Кроме того, задача имеет применения в химии и биологии [4–7], а также в силу своей сложности стала эталоном для тестирования численных методов оптимизации [8]. Стоит отметить, что в современных приложениях, например, в моделировании макромолекул [9] также применяются численные методы для минимизации потенциала. Конечно, часто потенциалы имеют более сложный вид и решается задача безусловной оптимизации, однако эти задачи имеют структуру, похожую на задачу Фекете в том смысле, что потенциал является суммой некоторых симметричных функций $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, что позволяет эффективно пересчитывать значение потенциала при смещении одной точки. В контексте вышеупомянутых приложений стоит обозначить, что целью работы является именно локальная минимизация функции (1), и в дальнейшем под \mathbf{x}^* понимается локальный минимум, ближайший к начальному положению \mathbf{x}^0 .

Учитывая широкое применение, задача Фекете была хорошо изучена [10–12]. Известны аналитические решения при $s = 1$ для $N = 2 - 6$ и 12. Для различных s и больших N получены решения с помощью численных методов [13]. В данной статье мы остановимся на обсуждении метода проективного покоординатного градиентного спуска, который использует быстрый счет функции и итеративный выбор шага, а также предложим алгоритм второго порядка для решения данной задачи.

Определение точек Фекете есть невыпуклая нелинейная задача оптимизации с нелинейными ограничениями. Следует отметить, что здесь можно легко избавиться от ограничений, параметризуя каждую точку на сфере двумя углами [14; 15], и решать задачу нелинейной оптимизации без ограничений от $2N$ переменных вместо $3N$. Однако на практике применение градиентных методов в таких координатах существенно замедляет счет оптимизируемой функции и градиента из-за относительно медленного вычисления тригонометрических функций. Поэтому остановимся подробнее на решении задачи в исходных координатах. Можно применять градиентные методы, рассматривая зависимость энергии от всех $3N$ переменных и на каждой итерации перемещать все точки одновременно. Сравнительный анализ в разделе 5 показывает, что покоординатный метод оказывается вполне конкурентоспособным. Более того, появляется возможность использовать покоординатную версию метода второго порядка, что оказывается затруднительным для полноградиентной версии (так как это требует подсчета и обращения матрицы размера $3N \times 3N$). Поэтому интересно рассматривать проективный покоординатный спуск в исходных координатах. Еще одним аргументом в пользу изучения покоординатного спуска на примере задачи вида (1) является то, что в последствии это дает возможность эффективно распараллелить вычисления. Если удастся эффективно организовать приближенный счет функции и его градиента по ближайшим соседям, то можно

на каждой итерации случайно разбивать множество точек так, чтобы в одну группу попадали близкие точки, и проводить блочно-покоординатный спуск параллельно для каждой группы [9]. В отличие от [16] в описанном ниже методе координата точки сразу после шага проектируется на поверхность сферы, а не раскладывается на нормальную и касательную составляющие.

2 Метод первого порядка

Зафиксируем положения $N - 1$ точек (без k -ой) на единичной сфере. Обозначим

$$U_k(y, \alpha_k) := U(\mathbf{x})|_{(x_1, \dots, x_{k-1}, y, x_{k+1}, \dots, x_N)} \quad (3)$$

функцию от k -ой переменной, в которой все остальные компоненты \mathbf{x} зафиксированы и являются параметрами $\alpha_k = (x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_N)$. Полуаем задачу относительно переменной ($x_k \in \mathbb{R}^3$) с ограничением на сфере:

$$y^* = \operatorname{argmin}_{y \in S^2} U_k(y, \alpha_k) \quad (4)$$

Оценим константу липшица градиента для (4). Градиент и гессиан можно вычислить аналитически, при $s > 0$:

$$\nabla U_k(y, \alpha_k) = - \sum_{i=1, i \neq k}^N \frac{s}{r_{ik}^{s+2}} (y - x_i) \quad (5)$$

При $s = 0$:

$$\nabla U_k(y, \alpha_k) = - \sum_{i=1, i \neq k}^N \frac{1}{r_{ik}^2} (y - x_i) \quad (6)$$

где $r_{ik} := \|y - x_i\|$ При $s > 0$:

$$\nabla^2 U_k(y, \alpha_k) = s \sum_{i=1, i \neq k} \frac{-I r_{ik}^2 + (s+2)(y - x_i)(y - x_i)^T}{r_{ik}^{s+4}} \quad (7)$$

При $s = 0$:

$$\nabla^2 U_k(y, \alpha_k) = \sum_{i=1, i \neq k} \frac{-I r_{ik}^2 + 2(y - x_i)(y - x_i)^T}{r_{ik}^4} \quad (8)$$

Пусть расстояние между каждой парой точек в \mathbf{x}^0 (заданная начальная конфигурация) больше некоторого $\epsilon > 0$. Тогда при $s > 0$:

$$\|\nabla^2 U_k(y, \alpha_k)\| \leq \frac{s(N-1)}{\epsilon^{s+2}} + \frac{s(s+2)(N-1)}{\epsilon^{s+2}} = \frac{s(s+3)(N-1)}{\epsilon^{s+2}} =: L \quad (9)$$

При $s = 0$:

$$\|\nabla^2 U_k(y, \alpha_k)\| \leq \frac{(N-1)}{\epsilon^{s+2}} + \frac{2(N-1)}{\epsilon^2} = \frac{3(N-1)}{\epsilon^2} =: L \quad (10)$$

Расстояние между любой парой точек на сфере можно ограничить снизу, если выбрать подходящее начальное приближение Раздел (4).

Тогда если обозначить $Q = \{\mathbf{x} \in \prod_{i=1}^N S_i^2 : U(\mathbf{x}) \leq U(\mathbf{x}^0)\}$, то на таком множестве будет выполнено:

$$\|\nabla U_k(x_k, \alpha_k) - \nabla U_k(y_k, \alpha_k)\| \leq L\|x_k - y_k\|, \forall k \in \overline{1, N}, \mathbf{x}, \mathbf{y} \in Q \quad (11)$$

Несложно показать, что минимизацией квадратичного приближения (4):

$$x_k^{n+1} = \underset{y \in S^2}{\operatorname{argmin}} \{U_k(x_k^n, \alpha_k^n) + \langle \nabla U_k(x_k^n, \alpha_k^n), y - x_k^n \rangle + \frac{L}{2} \|y - x_k^n\|^2\} \quad (12)$$

будет являться градиентный шаг с проектированием

$$x_k^{n+1} = \frac{x_k^n - h_n \nabla U_k(x_k^n, \alpha_k^n)}{\|x_k^n - h_n \nabla U_k(x_k^n, \alpha_k^n)\|} \quad (13)$$

где $h_n = \frac{1}{L}$, L - константа Липшица градиента.

В недавней работе [17] получены оценки сходимости для задачи (4) при некоторых предположениях. В частности, при условии (11) метод (13) генерирует невозрастающую по функции последовательность точек. Как следствие сформулируем покоординатную версию:

Утверждение 1 *Если для функции $U(\cdot)$ выполнено условие (11) для некоторого начального положения \mathbf{x}^0 . Тогда последовательность точек $\{\mathbf{x}^n\}_{n=1}^\infty$, полученная из (13) (например, последовательным обновлением компонент x_k), монотонно $U(\mathbf{x}^{n+1}) \leq U(\mathbf{x}^n)$, $\forall n \in \mathbb{N}$ сходится к стационарной точке функции $U(\cdot)$.*

Отметим, что для функции $U(\cdot)$, условие (11) не выполнено на всем множестве $\prod_{i=1}^N S_i^2$; однако, ясно, что если (9) выполнено для некоторого \mathbf{x} , то после шага (13) данная оценка не ухудшится и (11) будет справедливо на траектории.

Итак, опишем общую структуру метода:

1. Выбрать начальное приближение \mathbf{x}^0
2. Пока не достигнут критерий останова на n -ой итерации:
 - 2.1. Выбрать индекс k точки, которую будем двигать
 - 2.2. Вычислить шаг h_n
 - 2.3. Вычислить градиент $\nabla U_k(x_k^n, \alpha_k^n)$ относительно k -ой точки
 - 2.4. Сделать шаг (13), обновив k -ую компоненту \mathbf{x}^n

3 Метод второго порядка

Идея метода Ньютона и других методов второго порядка заключается в приближении оптимизируемой функции произвольной квадратичной функцией вместо параболоида вращения (12), что ведет к большему выигрышу по функции на каждой итерации. Однако несмотря на быструю сходимость таких алгоритмов по количеству итераций, их главным недостатком остается сложность вычисления и обращения матрицы вторых производных. В связи с чем активно развиваются различные модификации метода Ньютона, основанные на приближенном подсчете гессиана (квази-Ньютоновские методы) [18; 19], а также различные покоординатные версии [20; 21]. Здесь мы остановимся на покоординатной версии метода Ньютона для оптимизации с ограничением на сфере. Разложим функцию $U_k(z, \alpha_k) : \mathbb{R}^3 \rightarrow \mathbb{R}$ зависящую только от координаты одной точки до второго порядка в окрестности точки x_k :

$$\begin{aligned} & U_k(x_k, \alpha_k) + \nabla U_k(x_k, \alpha_k)(z - x_k) + \frac{1}{2}(\nabla^2 U_k(x_k, \alpha_k)(z - x_k), (z - x_k)) = \\ & = U_k(x_k, \alpha_k) - \nabla U_k(x_k, \alpha_k)x_k + \frac{1}{2}(\nabla^2 U_k(x_k, \alpha_k)x_k, x_k) + \nabla U_k(x_k, \alpha_k)z - \\ & - (\nabla^2 U_k(x_k, \alpha_k)x_k, z) + \frac{1}{2}(\nabla^2 U_k(x_k, \alpha_k)z, z) = \mathbf{C} + \frac{1}{2}(\mathbf{A}(\mathbf{x})z, z) - \mathbf{b}^T(\mathbf{x})z \end{aligned} \quad (14)$$

Здесь слагаемое $\mathbf{C} = U_k(x_k, \alpha_k) - \nabla U_k(x_k, \alpha_k)x_k + \frac{1}{2}(\nabla^2 U_k(x_k, \alpha_k)x_k, x_k)$ есть скалярная константа не зависящая от z . $\mathbf{A}(\mathbf{x}) = \nabla^2 U_k(x_k, \alpha_k)$. $\mathbf{b}(\mathbf{x}) = \nabla^2 U_k(x_k, \alpha_k)x_k - \nabla U_k(x_k, \alpha_k)$. Итак, мы получили вспомогательную задачу минимизации неоднородной квадратичной функции на сфере (15). Стоит заметить, что на каждой итерации гессиан $\mathbf{A}(\mathbf{x})$ не является положительно полуопределенным, так как существует направление (радиальное), вдоль которого функция возрастает вместе со своим квадратичным приближением. Отсюда следует, что бюджетное множество можно овыпуклить, заменив его единичным шаром, так как решение все равно будет лежать на поверхности сферы.

$$x_i^{n+1} = \operatorname{argmin}_{z \in S^2} \left\{ \frac{1}{2}z^T \mathbf{A}(\mathbf{x}^n)z - \mathbf{b}^T(\mathbf{x}^n)z \right\} \quad (15)$$

К сожалению, решение данной подзадачи нельзя получить аналогично (12, 13), то есть оно существенно зависит от структуры множества. Используя то, что в нашем случае множество - это единичная сфера, можно получить необходимые условия для решения данной задачи [22]. А именно:

Утверждение 2 Если $\lambda_1 < \lambda_2 \leq \dots \leq \lambda_m$ собственные числа матрицы \mathbf{A} , а ϕ_1, \dots, ϕ_m соответствующие ортонормированные собственные векторы и $\beta_i = \mathbf{b}^T(x)\phi_i$, $i = 1, \dots, m$, причем $\beta_1 \neq 0$. Тогда $\phi = \sum_{i=1}^m c_i \phi_i$ будет

решением задачи (15), если $c_i = \frac{\beta_i}{\lambda_i + \mu}$, $i = 1, \dots, m$, $\mu > -\lambda_1$ выбирается из условия:

$$\sum_{i=1}^m \frac{\beta_i^2}{(\lambda_i + \mu)^2} = 1 \quad (16)$$

Данный результат позволяет построить эффективный метод нахождения решения задачи (15), используя метод деления отрезка пополам. Для этого достаточно ограничить μ из утверждения (2). Поскольку $0 < \lambda_1 + \mu < \lambda_i + \mu$, $1 \leq i \leq m$, то

$$1 = \sum_{i=1}^m \frac{\beta_i^2}{(\lambda_i + \mu)^2} < \sum_{i=1}^m \frac{\beta_i^2}{(\lambda_1 + \mu)^2} = \frac{\|\mathbf{b}\|^2}{(\lambda_1 + \mu)^2}$$

С другой стороны:

$$1 = \sum_{i=1}^m \frac{\beta_i^2}{(\lambda_i + \mu)^2} > \frac{\beta_1^2}{(\lambda_1 + \mu)^2}$$

Объединяя две оценки, получим:

$$\mu_l = -\lambda_1 + \beta_1 \leq \mu \leq -\lambda_1 + \|\mathbf{b}\| = \mu_u$$

Итак, опишем общую структуру проективного покоординатного метода Ньютона:

1. Выбрать начальное приближение \mathbf{x}^0
2. Пока не достигнут критерий останова на n -ой итерации:
 - 2.1. Выбрать индекс k точки, которую будем двигать
 - 2.3. Вычислить градиент $\nabla U_k(x_k^n, \boldsymbol{\alpha}_k^n)$ и гессиан $\nabla^2 U_k(x_k^n, \boldsymbol{\alpha}_k^n)$
 - 2.3. Сделать шаг (15), обновив k -ую компоненту \mathbf{x}

4 Выбор начального приближения, шага, быстрый счет функции и градиента

4.1 Выбор начального приближения

Функции (1, 3), как уже было отмечено, являются невыпуклыми. Более того, эти функции разрывны на сфере и строго говоря никакие проективные градиентные методы для поиска минимума здесь не применимы. Однако, если выбор шага и начальной конфигурации гарантируют, что на траектории алгоритм не попадет в область разрывов, то можно получить гарантии сходимости. Итак, при применении градиентных методов для решения (2) важную роль играет выбор начального приближения. В литературе известно множество детерминированных [23; 24] и рандомизированных [25] алгоритмов расположения точек на поверхности сферы. Один из способов - метод Монте-Карло, в котором каждый элемент \mathbf{x}^0 выбирается из равномерного распределения на поверхности сферы. В предельном случае

$N \rightarrow \infty$ метод дает равномерную плотность распределения точек на сфере. Однако данный метод имеет явный недостаток в контексте выполнения условия (11), так как положение каждой точки является независимой случайной величиной, то минимальное расстояние между любой парой точек может быть сколь угодно малым, что, как легко видеть из (9, 10), это ведет к большой константе L и медленной сходимости. В работе [26] исследован метод "generalized spiral" и поведение $U(\mathbf{x}^0)$ в зависимости от количества точек N . Данный метод стремится расположить точки в узлах правильной сферической сетки из шестиугольников (hexagonal set). Так как данный детерминированный метод асимптотически ограничивает снизу минимальное расстояние между точками и дает разумное приближение задачи (2) по функции, ожидается, что условие (11) будет выполнено на множестве Q с небольшой константой L . А именно, с учетом оценок (9) и [27], получается при $s > 0$:

$$L = \frac{s(s+3)(N-1)(\sqrt{N})^{s+2}}{(\sqrt{4\pi})^{s+2}} \quad (17)$$

При $s = 0$:

$$L = \frac{3(N-1)N}{4\pi} \quad (18)$$

4.2 Быстрый счет функции и градиента

В силу специфики задачи можно существенно асимптотически ускорить каждую итерацию алгоритма по сравнению с наивным подходом с помощью быстрого подсчета функции. Как видно из (1) для подсчета энергии необходимо знать лишь попарные расстояния между точками. Поэтому будем хранить в памяти матрицу расстояний. Теперь на каждой итерации (для i -ой точки) будем обновлять энергию следующим образом: вычтем $N - 1$ слагаемое (те которые содержат индекс i) до выполнения шага, а затем прибавим соответствующие слагаемые после выполнения шага. Таким образом, если положить время вычисления одного слагаемого в (1) равным τ , то получим время вычисления порядка $2N\tau$ вместо $N^2\tau$.

Подсчет градиента на каждой итерации (т. е. для каждой точки) выполняется за линейное время по количеству точек N , поэтому суммарное время итерации также будет линейным. Здесь стоит отметить, что на каждой итерации мы будем хранить N градиентов вида (5, 6), упорядоченные по длине проекции на плоскость, касательную к сфере, то есть по критерию:

$$||(I - x_k x_k^T) \nabla U_k(x_k, \boldsymbol{\alpha}_k)|| \quad (19)$$

Последнее дает очень важную информацию о системе точек и позволяет определить какие точки находятся в неустойчивом положении, а значит дадут наибольшее уменьшение в энергии. При этом такой подход не будет затратным ни по времени ни по памяти. Казалось бы на каждой итерации придется считать N градиентов, что требует порядка N^2 операций, однако

данный подсчет можно провести за линейное время. Пусть на предыдущей итерации двигалась точка с индексом j , тогда для обновления градиентов (5) достаточно из каждого градиента $\nabla U_k(x_k, \alpha_k)$, $k \neq j$ вычесть $-\frac{s}{r_{jk}^{s+2}}(x_k - x_j)$ до выполнения итерации и прибавить такое же слагаемое после выполнения итерации. Кроме того, потребуется $O(N)$ операций, чтобы обновить градиент для j -ой точки. Согласно полученному критерию можно выбирать точки, например, с максимальной проекцией градиента или со взвешенными вероятностями, пропорциональными проекции градиента, причем такой выбор также делается за линейное время. Однако не стоит полагать, что выбор на каждой итерации точки с максимальным (19) будет всегда оптимальным с точки зрения поиска глобального минимума, потому что в таком случае есть большая вероятность сойтись к близкому, но не оптимальному локальному минимуму, что и наблюдается в экспериментах. Для локальной минимизации, мы используем именно такой подход, потому что он обеспечивает наилучшую релаксацию и более быструю сходимость.

4.3 Выбор шага

Так как константа липшица градиента (для k -ой координаты) L на множестве Q заранее неизвестна, то шаг h_n в (13) следует выбирать итеративно. Имея способ быстрого вычисления градиента, естественно выбирать шаг как величину обратную норме трехмерного гессиана в данной точке:

$$h_n = \frac{1}{L_n} = \frac{1}{\|\nabla^2 U_k(x_k, \alpha_k)\|} \quad (20)$$

Стоит отметить, что довольно скоро L_n перестает сильно меняться. Тогда здесь можно сэкономить время на подсчете гессиана и его нормы, если обновлять L_n не на каждой итерации.

5 Результаты вычислительных экспериментов

Для ясности дальнейшего анализа результатов, следуя [28], введем понятие "эпоха" (*epoch*) - N последовательных итераций покоординатного спуска, где N -количество точек. Итерацией по-прежнему будем называть шаг вида (13). При сравнении с полноградиентным методом естественно сопоставить один шаг полноградиентного метода с одной эпохой покоординатного спуска.

В разделе 4 предложен способ быстрого счета функции и градиента. Время выполнения метода первого порядка при фиксированном количестве итераций определяется линейными членами по количеству точек N . Мы исследовали зависимость реального времени выполнения метода (от количества точек N), запущенного на современном ноутбуке (процессор 2.8 GHz, оперативная память 16 ГБ). Как видно из Рис. 1 гипотеза о линейности оправдана.

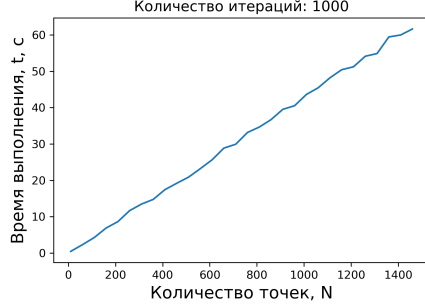


Рис. 1: Линейное время выполнения

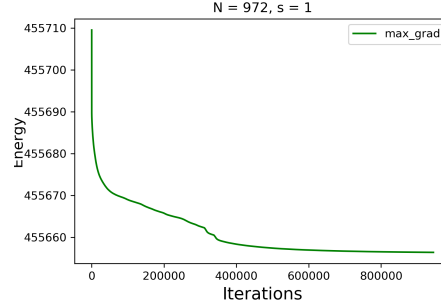


Рис. 2: Скорость сходимости, $N = 972, s = 1$

В предположении условия (11) было показано свойство релаксации. Интересно экспериментально выяснить скорость сходимости метода. Мы ожидаем получить сходимость по функции $U(\mathbf{x})$ со скоростью геометрической прогрессии с параметрами a и b :

$$U(\mathbf{x}^n) = aU(\mathbf{x}^{n-1}) + b \quad (21)$$

или

$$U(\mathbf{x}^n) = a^n U(\mathbf{x}^0) + b \frac{1 - a^{n-1}}{1 - a} \quad (22)$$

Для подбора параметров a и b используется нелинейный метод наименьших квадратов. График на Рис. 2 показывает скорость сходимости метода относительно функции (1) при $s = 1$ в режиме $epoch \leq N$. Это соответствует так называемому режиму "large scale" оптимизации, так как для данного покоординатного метода одна итерация оказывается в N раз дешевле подсчета полного градиента. Параметры МНК и минимально найденное значение функции $U(\mathbf{x}^n)$ для различных N указаны в таблице 1. В четвертом столбце рассчитано предельное значение аппроксимации $\frac{b}{1-a}$, к которому асимптотически стремится модель (22). Сравнивая столбцы $U(\mathbf{x}^n)$ и $U(\mathbf{x}^*)$ таблицы 1, можно заметить, что алгоритм достаточно близко приближается к минимально известным значениям. Различие порядка $10^{-3}\%$.

В разделе 4 также описывается возможность выбора точки для движения за время $O(N)$ на каждой итерации. На Рис. 3 изображены графики сходимости для различных стратегий при $s = 0$. Оказывается, что на начальном этапе стратегия выбора точки с максимальной проекцией градиента (*max_grad*) достаточно сильно обгоняет циклический перебор (*cyclic*). Стратегия (*max_grad_prob*), которая заключается в том, что точки выбираются пропорционально квадратам проекций, также оказывается достаточно хорошей. Вид графика оказывается типичным и при других s .

На Рис. 5 и Рис. 6 изображены графики сходимости на последних 100 из 5000 итераций для $N = 30$ для методов первого и второго порядка соот-

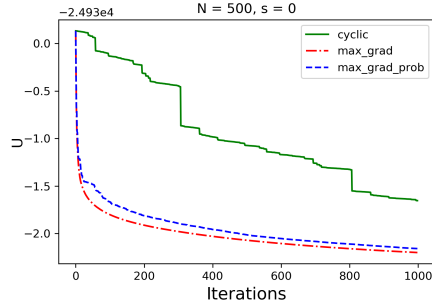


Рис. 3: Сравнение способов выбора точек для выполнения шага, $N = 500$, $s = 0$

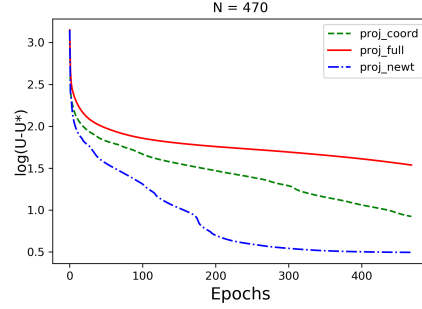


Рис. 4: Сравнение полноградиентного метода с покоординатными методами с градиентным и Ньютоновским шагами

Таблица 1: Параметры МНК при $s = 1$

N	a	U_{lim}	$U(\mathbf{x}^n)$	$U(\mathbf{x}^*)$	Итер.	Время, мин.
100	0.99981	4448.31	4448.49	4448.35	10000	0.6
470	0.99997	104826.19	104825.40	104822.89	220900	69.5
972	0.99997	455659.61	455656.38	455651.08	944784	650.5
1300	0.99998	819146.96	819143.51	-	1690000	1482

U_{lim} - предельное значение аппроксимации, $U(\mathbf{x}^n)$ - результат работы алгоритма, $U(\mathbf{x}^*)$ - минимально известное значение из литературы [13; 29]

ответственно. Начальные положения точек генерировались методом Монте-Карло. Можно заметить, что функция (1) для $s = 1$ является многоэкстремальной уже при $N = 30$. Интересно также, что метод второго порядка намного чаще достигает глобального минимума, а метод первого порядка застревает в точках, достаточно далеких от оптимума. Прерывистыми линиями выделены графики при генерации точек методом *generalized spiral*.

В разделе 3 предложен покоординатный аналог метода Ньютона с ограничением на сфере. Ожидается, что скорость сходимости по количеству итераций будет достаточно высокой, чтобы оправдать решение более сложной подзадачи на каждой итерации. Были проведены численные эксперименты с целью сравнить два предложенных метода в режиме $epoch \leq N$. На Рис. 4 изображены графики сходимости для данных двух методов и полноградиентного метода при $s = 1$. Оказывается, что время работы алгоритмов *proj_newt* и *proj_grad* в среднем отличаются примерно в два раза, при этом из графиков видно, что в рассмотренном режиме *proj_newt* обгоняет *proj_grad* больше, чем в два раза по функции. То есть, количество

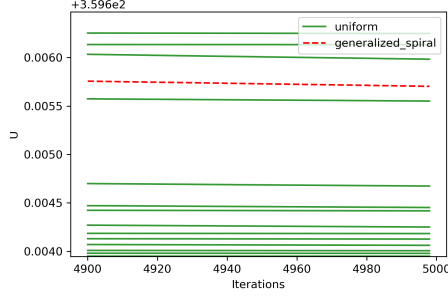


Рис. 5: Локальные минимумы для $N = 30, s = 1$, Метод 1 порядка

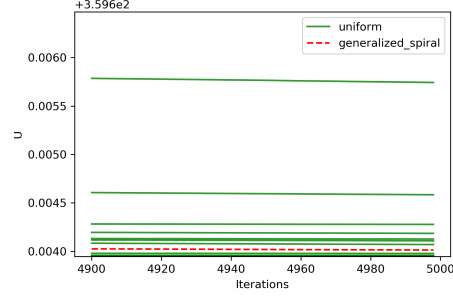


Рис. 6: Локальные минимумы для $N = 30, s = 1$, Метод 2 порядка

Таблица 2: Результаты экспериментов при $s = 1$ для больших N

N	$U(\mathbf{x}^*)$	Минимум (метод 1)	Минимум (метод 2)	Итераций (1, 2)	Время (1), мин.	Время (2), мин.
100	4448.351	4448.493	4448.450	10000	0.7	2.0
200	18438.843	18439.644	18439.156	40000	5.3	9.6
312	45629.314	45630.736	45629.926	97344	20	42
400	75582.449	75584.100	75583.854	160000	42	85
470	104822.886	104825.398	104824.527	220900	85	150

$U(\mathbf{x}^*)$ - минимально известное значение из литературы [13; 29]

итераций, которое требуется методу *proj_newt*, чтобы достичь значения функции, которое достиг *proj_grad* за N эпох меньше, чем $N/2$. Также можно заметить, что оба метода значительно обгоняют полноградиентный метод при рассмотрении сходимости по функции. В таблицах 2 и 3 приведены результаты работы двух методов. В таблице 3 критерием останова кроме количества итераций задано изменение по функции на последних двух эпохах (если оно оказывается меньше 10^{-12} , то алгоритм останавливается). Можно заметить, что метод 2 порядка (*proj_newt*) значительно чаще достигает меньших минимумов, при этом ему часто для этого требуется меньшее число итераций.

Расположение точек на сфере независимо от вида задачи (1) наглядно представляются с помощью сферических диаграмм Вороного. Данные диаграммы представляют собой разбиение поверхности сферы на области (многоугольники), в которых каждая точка является ближайшей к данной точке (электрону). На Рис. 7-10 представлены диаграммы Вороного, в каждой паре слева - диаграмма при начальной генерации точек методом *generalizedspiral*, справа - диаграмма после оптимизации с функцией (1) при $s = 1$. Отсюда можно заметить, что выбранное начальное приближение располагает точки достаточно равномерно, структура диаграмм после оптимизации согласуются с результатами из литературы [11; 29].

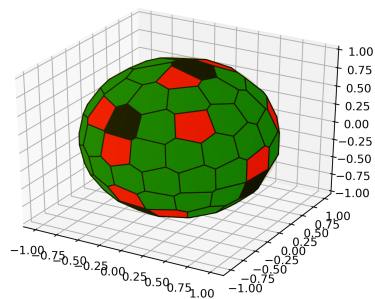


Рис. 7: Диаграмма Вороного до оптимизации, $N = 100$

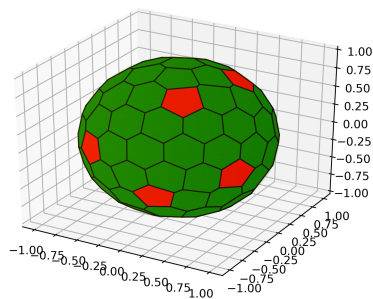


Рис. 8: Диаграмма Вороного после оптимизации, $N = 100$

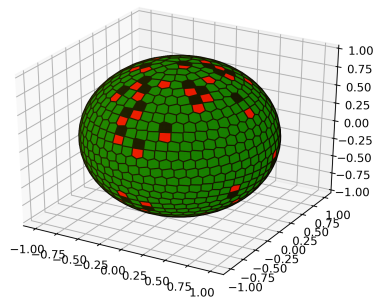


Рис. 9: Диаграмма Вороного до оптимизации, $N = 972$

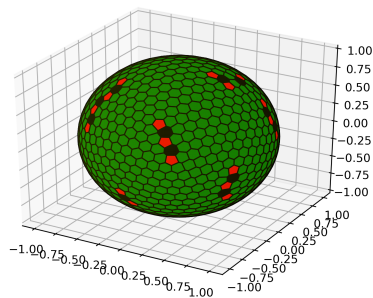


Рис. 10: Диаграмма Вороного после оптимизации, $N = 972$

6 Заключение

Предложены два проективных покоординатных метода для поиска локального минимума в задаче Фекете. Алгоритмы используют быстрый счет функции, градиента и выбор наилучшей компоненты для оптимизации. Наши численные эксперименты показывают превосходство предложенных методов перед полноградиентным методом при количестве точек $N < 1000$. Можно ожидать, что предложенные методы окажутся предпочтительными для задач больших размерностей.

Список литературы

1. *Fekete M.* Über die Verteilung der Wurzeln bei gewissen algebraischen Gleichungen mit ganzzahligen Koeffizienten // *Mathematische Zeitschrift*. — 1923. — Т. 17. — С. 228—249.
2. *Thomson J. J.* On the structure of the atom: an investigation of the stability and periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure // *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*. — 1904. — Т. 7, № 39. — С. 237—265.
3. *Smale S.* Mathematical problems for the next century // *The Mathematical Intelligencer*. — 1998. — Т. 20, № 2. — С. 7—15.
4. *Robinson M., Suarez-Martinez I., Marks N. A.* Generalized method for constructing the atomic coordinates of nanotube caps // *Phys. Rev. B*. — 2013. — Т. 87, вып. 15. — С. 155430.
5. *Patra M., Patriarca M., Karttunen M.* Stability of charge inversion, Thomson problem, and application to electrophoresis // *Physical review. E, Statistical, nonlinear, and soft matter physics*. — 2003. — Т. 67. — С. 031402.
6. *Caspar D., Klug A.* Physical Principles in the Construction of Regular Viruses // *Cold Spring Harbor symposia on quantitative biology*. — 1962. — Т. 27. — С. 1—24.
7. *Arias E., E. F., Pérez-Torres J. F.* Algorithm based on the Thomson problem for determination of equilibrium structures of metal nanoclusters // *J. Chem. Phys.* — 2017.
8. Generalized Simulated Annealing for Global Optimization: The GenSA Package / Y. Xiang [и др.] // *The R Journal*. — 2013. — Т. 5.
9. Алгоритмы локальной минимизации силового поля для трехмерных макромолекул / П. А. Яковлев [и др.] // *arXiv preprint arXiv:1810.03358v2*. — 2018.
10. *Altschuler E. L., Pérez-Garrido A.* New Global Minima for Thomson's Problem of Charges on a Sphere. — 2004.

11. *Wales D. J., McKay H., Altschuler E. L.* Defect motifs for spherical topologies // *Phys. Rev. B.* — 2009. — Т. 79, вып. 22. — С. 224115.
12. *LaFave Jr T.* Discrete Transformations in the Thomson Problem // *Journal of Electrostatics.* — 2014. — Т. 72. — С. 39–43.
13. *Ridgway J. W., Cheviakov A. F.* An iterative procedure for finding locally and globally optimal arrangements of particles on the unit sphere // *Computer Physics Communications.* — 2018. — Т. 233.
14. *Lakhabab H., EL Bernoussi S., EL Harif A.* Energy Minimization of Point Charges on a Sphere with a Spectral Projected Gradient Method // *International Journal of Scientific & Engineering Research.* — 2012. — Т. 3.
15. *J. Nurmela K.* Constructing Spherical Codes by Global Optimization Methods. — 1995.
16. Estimation of Fekete points / E. Bendito [и др.] // *Journal of Computational Physics.* — 2007. — Т. 225, № 2. — С. 2354–2376.
17. *Polyak B. T., Balashov M. V., Tremba A. A.* The gradient projection algorithm and the conditional gradient method for minimization on nonconvex sets and surfaces. — 2019.
18. *Liu D. C., Nocedal J.* On the limited memory BFGS method for large scale optimization // *Mathematical Programming.* — 1989. — Т. 45, № 1. — С. 503–528.
19. A Stochastic Quasi-Newton Method for Large-Scale Optimization / H. R. Byrd [и др.] // *SIAM Journal on Optimization.* — 2014. — Т. 26.
20. *Mutny M., Richtarik P.* Parallel stochastic Newton method // *Journal of Computational Mathematics.* — 2018. — Т. 36, № 3. — С. 404.
21. *Doikov N., Richtarik P.* Randomized Block Cubic Newton Method. — 2018.
22. *Hager W.* Minimizing a Quadratic Over a Sphere // *SIAM Journal on Optimization.* — 2001. — Т. 12, № 1. — С. 188–208.
23. *Yershova A., M. LaValle S.* Deterministic Sampling Methods for Spheres and SO(3) // *Proceedings - IEEE International Conference on Robotics and Automation.* — 2003. — Т. 2004.
24. *Saff E. B., Kuijlaars A. B. J.* Distributing many points on a sphere // *The Mathematical Intelligencer.* — 1997. — Т. 19, № 1. — С. 5–11.
25. *Yershova A., LaValle S. M., Mitchell J. C.* Generating Uniform Incremental Grids on SO(3) Using the Hopf Fibration // *Algorithmic Foundation of Robotics VIII: Selected Contributions of the Eight International Workshop on the Algorithmic Foundations of Robotics* / под ред. G. S. Chirikjian [и др.]. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2010. — С. 385–399.
26. *Rakhmanov E., Saff B. E., Zhou M. Y.* Minimal Discrete Energy on the Sphere // *Mathematical Research Letters.* — 1994. — Т. 1. — С. 647–662.

27. *Hardin D., Michaels T., E.B. S.* A Comparison of Popular Point Configurations on S^2 // Dolomites Research Notes on Approximation. — 2016. — T. 9. — C. 16—49.
28. *Wright S. J.* Coordinate descent algorithms // Mathematical Programming. — 2015. — T. 151, № 1. — C. 3—34.
29. *Wales D., Ulker S.* Structure and dynamics of spherical crystals characterized for the Thomson problem // Phys. Rev. B. — 2006. — T. 74.

Таблица 3: Результаты экспериментов при $s = 1$ для небольших N

N	$U(\mathbf{x}^*)$	Минимум (метод 1)	Минимум (метод 2)	Итераций (1)	Итераций (2)	Время (1), сек.	Время (2), сек.
2	0.500000000	0.500000000	0.500000000	56	9	0.01	0.01
3	1.732050808	1.732050808	1.732050808	81	29	0.02	0.06
4	3.674234614	3.674234614	3.674234614	94	36	0.02	0.08
5	6.474691495	6.474691495	6.474691495	379	74	0.10	0.17
6	9.985281374	9.985281374	9.985281374	287	90	0.09	0.17
7	14.452977414	14.452977417	14.452977414	7350	1911	2.46	6.02
8	19.675287861	19.675287861	19.675287861	1455	439	0.54	1.37
9	25.759986531	25.759986531	25.759986531	2256	727	0.94	2.17
10	32.716949460	32.716949460	32.716949460	2827	1389	1.30	4.77
11	40.596450510	40.596450508	40.596450508	3439	1494	1.73	5.08
12	49.165253058	49.165253058	49.165253058	630	353	0.35	0.99
13	58.853230612	58.853230612	58.853230612	7719	5354	4.52	20.54
14	69.306363297	69.306363297	69.306363297	1905	1266	1.21	4.39
15	80.670244114	80.670244114	80.670244114	3821	2066	2.56	7.76
16	92.911655302	92.920353962	92.920353962	2670	1981	1.91	8.92
17	106.050404829	106.050404829	106.050404829	6335	4067	4.77	16.18
18	120.084467447	120.084467447	120.084467447	4514	3711	3.59	17.08
19	135.089467557	135.089467598	135.089467557	54150	54150	46.95	282.30
20	150.881568334	150.881568334	150.881568334	9403	5335	8.46	24.17
21	167.641622399	167.641622399	167.641622399	32935	22896	31.23	148.31
22	185.287536149	185.287536149	185.287536149	6076	4306	5.96	23.12
23	203.930190663	203.930190663	203.930190663	5383	3439	5.54	16.05
24	223.347074052	223.347074052	223.347074052	7426	4837	7.92	24.45
25	243.812760299	243.812760306	243.812760299	93750	76103	108.55	442.88
26	265.133326317	265.133326317	265.133326317	45109	27332	52.59	189.53
27	287.302615033	287.302615033	287.302615033	12481	7414	14.98	40.95
28	310.491542358	310.491542358	310.491542358	8950	5935	11.06	33.30
29	334.634439920	334.634439921	334.634439920	44380	28592	57.22	195.81
30	359.603945904	359.603945904	359.603945904	28068	24989	37.12	148.93
31	385.530838063	385.530838063	385.530838063	9762	4572	13.31	29.12
32	412.261274651	412.261274651	412.261274651	5476	3136	7.76	21.35
33	440.204057448	440.204057449	440.204057448	163350	117552	255.54	809.63
34	468.904853281	468.904853281	468.904853281	26628	14157	40.05	93.49
35	498.569872491	498.573454038	498.573454038	183750	97072	309.04	796.23
36	529.122408375	529.122408762	529.122408475	194400	194400	339.14	1710.69
37	560.618887731	560.618887731	560.618887731	39055	19634	63.97	161.08
38	593.038503566	593.038503566	593.038503566	12768	6769	21.46	54.07
39	626.389009017	626.389009017	626.389009017	12929	6634	22.17	48.01
40	660.675278835	660.675278835	660.675278835	15831	8542	26.93	62.92
41	695.916744342	695.916744342	695.916744342	12799	6691	22.22	52.37
42	732.078107544	732.078107544	732.078107544	26700	14488	47.95	98.29
43	769.190846459	769.190846460	769.190846460	213084	95671	436.80	959.87
44	807.174263085	807.174263085	807.174263085	112652	40445	220.13	400.13
45	846.188401061	846.188401062	846.188401062	303750	85123	711.34	870.44
46	886.167113639	886.170216023	886.171432425	38313	26262	74.65	210.60
47	927.059270680	927.072224565	927.072224565	164854	65438	357.04	363.51
48	968.713455344	968.713455344	968.713455344	34190	16851	72.21	152.74
49	1011.557182654	1011.557182654	1011.557182654	109322	30728	242.43	232.27
50	1055.182314726	1055.182314726	1055.182314726	32643	15879	72.45	136.42

$U(\mathbf{x}^*)$ - минимально известное значение из литературы [13; 29]

Сложность задачи (2) оказывается не монотонной с точки зрения требуемого количества итераций для сходимости по функции. Так, например, при $N = 19, 36$ обоим методам требуется значительное количество итераций, при том что задачи при $N = 20, 37$ решаются намного быстрее.