Применение проективного покоординатного спуска в задаче Фекете

Поляк Б. Т. ИПУ РАН, Москва
Фатхуллин И. Ф. МФТИ, Москва

14 апреля 2019 г.

Аннотация

Рассматривается задача минимизации энергии системы из N точек, с ограничением на поверхности сферы в $\mathbb{R}^3$, и взаимодействующих с потенциалом $U = \frac{1}{r^s}, s > 0$, где r - евклидово расстояние между парой точек. Сложность подсчета функции и градиента в данной задаче оказываются квадратичными по числу переменных. 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. Найти множество точек $\omega_N = \{x_1, \ldots, x_N\}$ 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:

$$E(\omega_N) := \sum_{1 \leq i < j \leq N} U(x_i, x_j) \tag{1}$$

То есть формально решается задача:

$$\hat{\omega}_N = \underset{\omega_N \in \prod_{i=1}^N S_i^2}{\operatorname{argmin}} E(\omega_N), \tag{2}$$

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 $r_{ij} = ||x_i - x_j||_2$: $\widetilde{U}(r_{ij}) = U(x_i, x_j)$. Особый интерес представляет изучение ядерной функции Рисса $\widetilde{U}(r_{ij}) = r_{ij}^{-s}$, $s > 0$ и логарифмического ядра $\widetilde{U}(r_{ij}) = -log(r_{ij})$.

При $s = 1$ получается Кулоновский потенциал и соответственно задача Томсона. Последняя была поставлена Дж. Дж. Томсоном еще в 1904 году, однако до сих пор остается нерешенной для произвольного N. Задача Фекете привлекла особое внимание математиков, потому что она появляется в списке Смейла [**Smale1998**] - нерешенных задач для 21-го века. Кроме того, задача имеет применения в химии и биологии [**PhysRevB.87.155430**; **Stability**; **Viruses**; **Nanoclusters**], а также в силу своей сложности стала эталоном для тестирования численных методов оптимизации [**Sim_Annealing**]. Стоит отметить, что в современных приложениях, например, в моделировании макромолекул [**3D_macro_molecule**] 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 $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, что позволяет эффективно пересчитывать значение потенциала при смещении одной точки.

Учитывая широкое применение, задача Фекете была хорошо изучена [**New_Minima**; **Def_motifs**; **Discr_Transf**]. Известны аналитические решения при $s = 1$ для $N = 2{-}6$ и 12. Для различных s и больших N получены решения с помощью численных методов [**Iterative_procedure_table_s**]. 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 [**Spectral_projected**; **Codes**], 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 было бы применять градиентные методы, рассматривая зависимость энергии от всех $3N$ переменных и на каждой итерации перемещать все точки. Однако в таком подходе уже при $N \sim 10^4$ (особенно в случае использования методов второго порядка) у современного персонального компьютера начинаются проблемы с памятью. 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 удается эффективно организовать приближенный счет функции и его градиента по ближайшим соседям, то можно на каждой итерации случайно разбивать множество точек ω_N так, чтобы в одну группу попадали близкие точки, и проводить покоординатный спуск параллельно для каждой группы [**3D_macro_molecule**]. В отличие от

[**Estimation_Fekete**] в описанном ниже методе координата точки сразу после шага проектируется на поверхность сферы, а не раскладывается на нормальную и касательную составляющие.

2 Выбор начального приближения, быстрый счет функции и градиента

Выбор начального приближения играет важную роль в сходимости градиентных методов. В литературе известно множество детерминированных [**Det_Sampling**; **Saff1997**] и рандомизированных [**Yershova2010**] алгоритмов расположения точек на поверхности сферы. Один из способов - метод Монте-Карло, в котором каждый элемент ω_N^0 выбирается из равномерного распределения на поверхности сферы. В предельном случае $N \to \infty$ метод дает равномерную плотность распределения точек на сфере. Однако данный метод имеет явный недостаток в контексте выполнения условия (**??**), 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, ведет к большой константе L и медленной сходимости. В работе [**Saff**] исследован метод "generalized spiral"и поведение $E(\omega_N^0)$ в зависимости от количества точек N. Данный метод стремится расположить точки в узлах правильной сферической сетки из шестиугольников (hexogonal set). Так как данный детерминированный метод ограничивает снизу минимальное расстояние между точками и дает разумное приближение задачи (2) по функции, ожидается, что условие (10) будет выполнено на множестве Q.

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 $N^2\tau$.

Градиент, функции можно вычислить аналитически:

$$\nabla \widetilde{E}(x_k) = -\sum_{i=1, i\neq k}^{N} \frac{s}{r_{ik}^{s+2}}(x_k - x_i) \tag{3}$$

где $r_{ik} := ||x_i - x_k||$ 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 (3), упорядоченные по длине проекции на плоскость, касательную к сфере, то есть

по критерию:

$$||(I - x_k x_k^T)\nabla \widetilde{E}(x_k)|| \tag{4}$$

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 линенейное время. Пусть на предыдущей итерации двигалась точка с индексом j, тогда для обновления градиентов (3) достаточно из каждого градиента $\nabla \widetilde{E}(x_k)$, $k \neq j$ 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, выбирать точку 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 (4) 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. Впрочем все зависит от целей, ведь иногда важно быстро сойтись к локальному минимуму [**3D_macro_molecule**].

3 Метод первого порядка

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

$$\widetilde{E}(y) = \widetilde{E_i}(y) := E(s, \omega_N)|_{\{x_1^0, \ldots, x_{i-1}^0, y, x_{i+1}^0, \ldots, x_N^0\}} \tag{5}$$

где y стоит на i-ом месте, а все остальные компоненты ω_N зафиксированы. Получим задачу относительно трех переменных ($x_i \in \mathbb{R}^3$) с ограничением на сфере:

$$x_i^{n+1} = \operatorname*{argmin}_{y \in S^2} \widetilde{E}(y) \tag{6}$$

Тогда в предположении липшицевости градиента функции $\widetilde{E}(\cdot)$

$$\nabla \widetilde{E}(x) - \nabla \widetilde{E}(y) \leq L||x - y||, \forall x, y \in S^2 \tag{7}$$

Можно показать, что градиентный шаг с проектированием (8) минимизирует квадратичное приближение функции.

$$x_i^{n+1} = \frac{x_i^n - \alpha_n \nabla \widetilde{E}(x_i^n)}{||x_i^n - \alpha_n \nabla \widetilde{E}(x_i^n)||} \tag{8}$$

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

Стоит отметить, что данное утверждение остается верным даже в случае произвольного ограничения $\mathfrak{Q}$, нужно лишь правильно проектировать на заданное множество:

$$x_i^{n+1} = arg\min_{x\in\mathfrak{Q}}\{\widetilde{E}(x_i^n) + < \nabla\widetilde{E}(x_i^n), x - x_i^n > + \frac{L}{2}||x - x_i^n||_2^2\} =$$
$$= arg\min_{x\in\mathfrak{Q}}\{2 < \frac{1}{L}\nabla\widetilde{E}(x_i^n), x - x_i^n > + ||x - x_i^n||_2^2 + \frac{1}{L^2}||\nabla\widetilde{E}(x_i^n)||_2^2\} =$$
$$= arg\min_{x\in\mathfrak{Q}}\{||x - (x_i^n - \frac{1}{L}\nabla\widetilde{E}(x_i^n))||_2^2\} = \pi_{\mathfrak{Q}}\left(x_i^n - \frac{1}{L}\nabla\widetilde{E}(x_i^n)\right) \qquad (9)$$

В недавней работе [**Polyak**] получены оценки сходимости для задачи (6) при некоторых предположениях. А именно, при условии (7) метод (8) генерирует невозрастающую по функции последовательность точек, а при дополнительном условии (10)

$$||(I - xx^T)\nabla\widetilde{E}(x)||^2 \geq \mu(\widetilde{E}(x) - \widetilde{E}^*), \forall x \in S^2 \qquad (10)$$

последовательность сходится к минимуму $x^* : \widetilde{E}(x^*) = \widetilde{E}^*$ с линейной скоростью как по функции, так и по аргументу. Условие (10) лишь требует, чтобы проекция градиента на касательное подпространство возрастала достаточно быстро при отклонении от минимума и, как следствие, чтобы всякая стационарная точка была точкой минимума. Стоит также отметить, что из (10) не следует единственность минимума. Как следствие теорем 1 и 2 из [**Polyak**], можно получить сходимость покоординатного метода со скоростью геометрической прогрессии:

Теорема 1 *Пусть функция $E(\cdot)$ такова, что для каждого $i = \overline{1,N}$ функция $\widetilde{E}_i(\cdot)$, определенная в (5) имеет липшицев градиент (т. е. выполнено условие (7)) с константами $L_1, \ldots, L_N$. Тогда последовательность точек $\{\omega_N^n\}_{n=1}^{\infty}$, полученная из (8) (например, последовательным перебором точек x_i), монотонно $E(\omega_N^{n+1}) \leq E(\omega_N^n), \forall n \in \mathbb{N}$ сходится к стационарной точке функции $E(\cdot)$.*

Если дополнительно известно, что для функций $\{\widetilde{E}_i(\cdot)\}_{i=1}^N$ выполнено условие (10) с константами $\mu_1, \ldots, \mu_N$ и липшецевость с константами $M_1, \ldots, M_N$ (где $\widetilde{E}^* := E^* = E(\omega_N^*)$: ω_N^* минимум функции $E(\cdot)$), то последовательность $\{\omega_N^n\}_{n=1}^{\infty}$ сходится к локальному минимуму с линейной скоростью.

Причем $E(\omega_N^{k+1}) - E^* \leq \left(1 - \frac{\min\limits_{i=\overline{1,N}}\mu_i}{\max\limits_{i=\overline{1,N}}L_i + \max\limits_{i=\overline{1,N}}M_i}\right)^k (E(\omega_N^0) - E^*)$

В полученной оценке, к сожалению, не удается строго оценить покоординатные константы μ_i, L_i, M_i, соответствующими константами для функции $E(\cdot)$. Поэтому пользуясь специфической симметричностью задачи (1) относительно переменных $x_1, \ldots, x_N$, а также полагаясь на хорошее начальное

приближение будем считать, что $\mu = N\mu_i, L = NL_i, M = NM_i$. При таком предположении, N условий вида (10) можно ослабить, введя одно условие для функции $E(\cdot)$:

$$||B\nabla E(\omega_N)||^2 \geq \mu(E(\omega_N) - E^*) \tag{11}$$

где $B \in Mat(3N \times 3N)$ - блочно-диагональная матрица с диагональными элементами вида $I - x_i x_i^T, i \in \overline{1,N}, x_i \in \mathbb{R}^3$. Тогда получается оценка:

$$E(\omega_N^{k+1}) - E^* \leq \left(1 - \frac{\mu}{L+M}\right)^k (E(\omega_N^0) - E^*) \tag{12}$$

Отметим, что для функции $E(\cdot)$, вообще говоря, условие (7) не выполнено. Однако если удается удачно выбрать начальное приближение ω_N^0, то данное условие выполняется на множестве $Q = \{\omega_N \in \prod_{i=1}^N S_i^2 : E(\omega_N) \leq E(\omega_N^0)\}$ с небольшой константой L.

Итак, опишем здесь общую структуру метода, а в следующих разделах остановимся подробнее на отдельных моментах.

1. Выбрать начальное приближение ω_N^0
2. Пока не достигнут критерий останова:
2.1. Вычислить шаг α^n
2.2. Выбрать индекс i точки, которую будем двигать
2.3. Вычислить градиент относительно i -ой точки $\nabla\widetilde{E}(x_i)$
2.4. Сделать шаг (8), обновив i-ую компоненту ω_N^n

4 Метод второго порядка

Идея метода Ньютона и других методов второго порядка заключается в приближении оптимизируемой функции произвольной квадратичной функцией вместо параболоида вращения (9), что ведет к большему выигрышу по функции на каждой итерации. Однако несмотря на быструю сходимость таких алгоритмов по количеству итераций, их главным недостатком остается вычислительная 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) [**BFGS_Liu1989**; **Quasi_Newton_2014**], а также различные покоординатные версии [**PSN_Richtarik**; **RBCN_Richtarik**]. Здесь мы остановимся на покоординатной версии метода Ньютона для оптимизации с ограничением на сфере. Если в способе выбора шага уже вычисляется гессиан от трех переменных, то можно использовать методы второго порядка без дополнительного обращения к оракулу и ожидать более быструю сходимость. Разложим функцию $\widetilde{E} : \mathbb{R}^3 \to \mathbb{R}$ зависящую только от координаты одной точки до второго порядка в окрестности точки x:

$$\widetilde{E}(z) = \widetilde{E}(x) + \nabla\widetilde{E}(x)(z-x) + \frac{1}{2}(\nabla^2\widetilde{E}(x)(z-x),(z-x)) =$$
$$= \widetilde{E}(x) - \nabla\widetilde{E}(x)x + \frac{1}{2}(\nabla^2\widetilde{E}(x)x,x) + \nabla\widetilde{E}(x)z - (\nabla^2\widetilde{E}(x)x,z) +$$
$$+ \frac{1}{2}(\nabla^2\widetilde{E}(x)z,z) = \mathbf{C} + \frac{1}{2}(\mathbf{A}(x)z,z) - \mathbf{b}^T(x)z \tag{13}$$

Здесь слагаемое $\mathbf{C} = \widetilde{E}(x) - \nabla\widetilde{E}(x)x + \frac{1}{2}(\nabla^2\widetilde{E}(x)x,x)$ есть скалярная константа не зависящая от z. $\mathbf{A}(x) = \nabla^2\widetilde{E}(x)$. $\mathbf{b}(x) = \frac{1}{2}(\nabla^2\widetilde{E}(x)x - \nabla\widetilde{E}(x))$. Итак, мы получили вспомогательную задачу минимизации неоднородной квадратичной функции на сфере (14). Стоит заметить, что на каждой итерации гессиан $\mathbf{A}(x)$ 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.

$$x_i^{n+1} = \underset{z\in S^2}{\operatorname{argmin}}\{\frac{1}{2}z^T\mathbf{A}(x_i^n)z - \mathbf{b}^T(x_i^n)z\} \tag{14}$$

К сожалению, решение данной подзадачи нельзя получить аналогично (9), то есть оно существенно зависит от структуры множества $\mathfrak{Q}$. Используя то, что в нашем случае $\mathfrak{Q} = S^2$ можно получить необходимые условия для решения данной задачи [**Min_quad**]. А именно:

Теорема 2 *Если $\lambda_1 < \lambda_2 \leq \cdots \leq \lambda_m$ собственные числа матрицы $\mathbf{A}$, а $\phi_1,\ldots,\phi_m$ соответствующие ортонормированные собственные векторы и $\beta_i = \boldsymbol{b}^T(x)\phi_i$, $i = 1,\ldots,m$, причем $\beta_1 \neq 0$. Тогда $\phi = \sum_{i=1}^m c_i\phi_i$ будет решением задачи (14), если $c_i = \frac{\beta_i}{\lambda_i+\mu}$, $i = 1,\ldots,m$, $\mu > -\lambda_1$ выбирается из условия:*

$$\sum_{i=1}^m \frac{\beta_i^2}{(\lambda_i+\mu)^2} = 1 \tag{15}$$

Данный результат позволяет построить эффективный метод нахождения решения задачи (14), используя метод деления отрезка пополам. Для этого достаточно ограничить μ из теоремы (2). Следуя работе [**Min_quad**], можно получить:

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

5 Результаты вычислительных экспериментов

В разделе 2 предложен способ быстрого счета функции и градиента. Время выполнения метода первого порядка при фиксированном количестве ите-

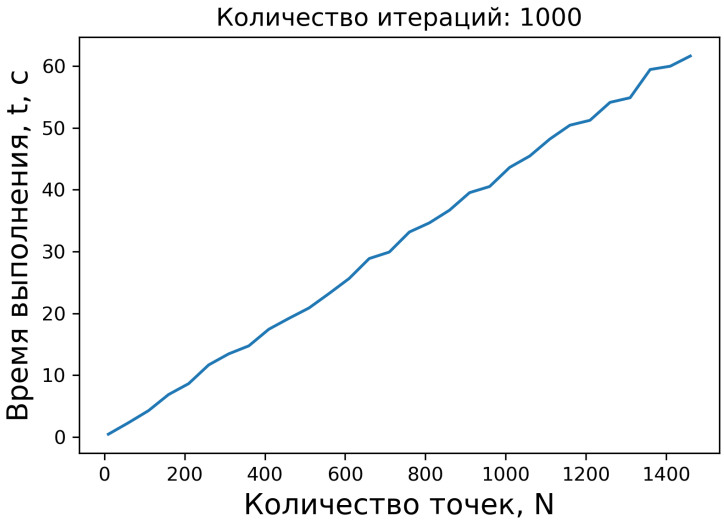

Рис. 1: линейное время выполнения

раций 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.

В предположении условий (7) и (10) было получено условие линейной сходимости по функции для описанного метода первого порядка с последовательным выбором точек. То есть мы ожидаем получить сходимость оптимизируемой функции энергии E со скоростью геометрической прогресии с параметрами a и b:

$$E(n) = aE(n-1) + b \tag{16}$$

или

$$E(n) = a^n E(0) + b\frac{1-a^{k-1}}{1-a} \tag{17}$$

Для подбора параметров a и b используется нелинейный метод наименьших квадратов. Графики на Рис. 2, 3 показывают скорость сходимости метода относительно функции (1) при $s = 1$ в режиме "количество итераций"$\leq N^2$. 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 E для различных N указаны в таблице 1. В четвертом столбце рассчитано предельное значение аппроксимации $\frac{b}{1-a}$, к которому ассимптотически стремится модель (17).

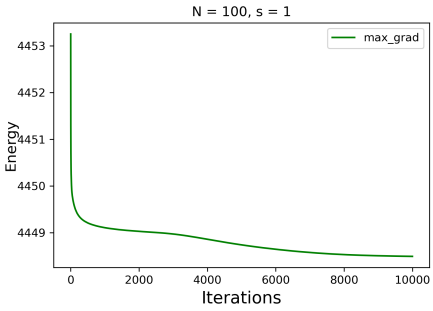

Рис. 2: скорость сходимости

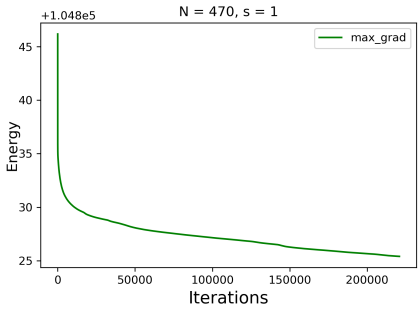

Рис. 3: скорость сходимости

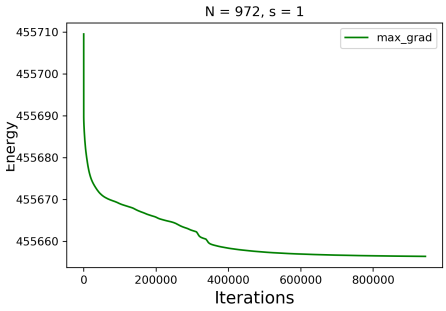

Рис. 4: скорость сходимости

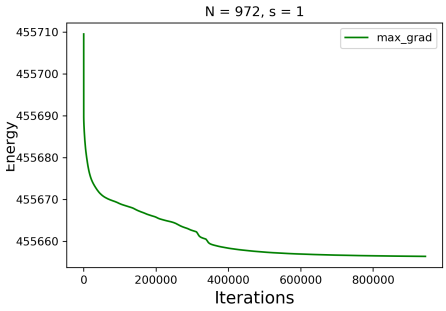

Рис. 5: скорость сходимости

Сравнивая столбцы $E(N)$ и E_{min} таблицы 1, можно заметить, что алгоритм достаточно близко приближается к минимально известным значениям. Различие: $10^{-4} - 10^{-2}\%$.

В разделе 2 также описывается возможность выбора точки для движения за время $O(N)$. На Рис. 6, 7, 8, 9 изображены графики сходимости для различных стратегий и для различных s. 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.

На Рис. 10 и Рис. 11 изображены графики сходимости на последних 100 из 500 итераций для $N = 10$ для методов первого и второго порядка соответственно. Начальные положения точек генерировались методом Монте-Карло. Можно заметить, что задача (2) является многоэкстремальной уже при $N = 10$. Интересно также, что метод второго порядка намного чаще достигает глобального минимума, а метод первого порядка застревает в точках, достаточно далеких от оптимума. Красными линиями выделены графики при генерации точек методом generalized spiral.

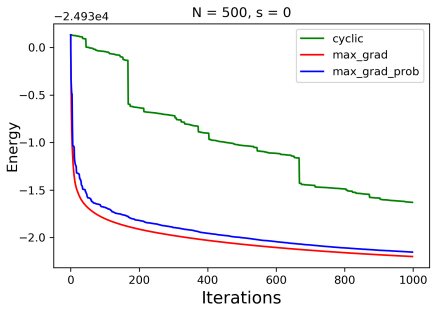

Рис. 6: Выбор точек для выполнения шага

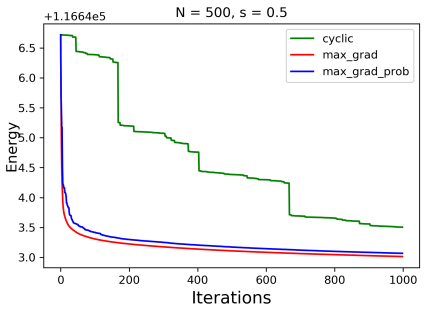

Рис. 7: Выбор точек для выполнения шага

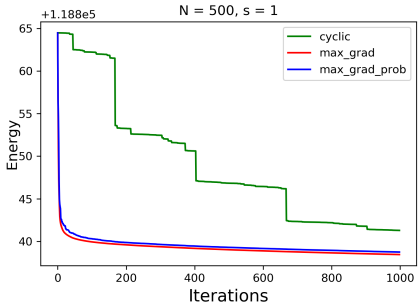

Рис. 8: Выбор точек для выполнения шага

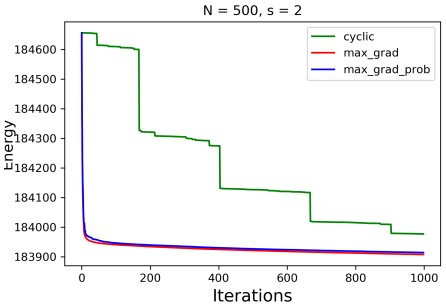

Рис. 9: Выбор точек для выполнения шага

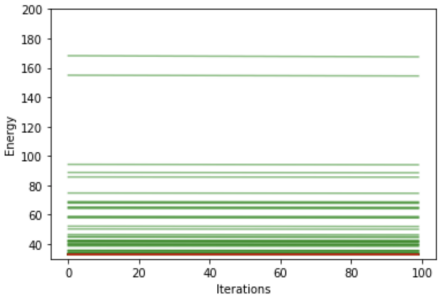

Рис. 10: Метод 1 порядка

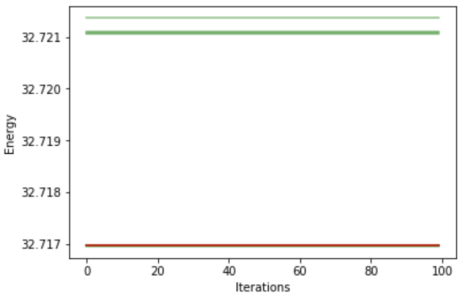

Рис. 11: Метод 2 порядка

Таблица 1: Параметры МНК при $s = 1$

N	a	b	$\frac{b}{1-a}$	$E(N)$	E_{min}	Итер.	Время, мин.
100	0.99981	0.85	4448.31	4448.49	4448.35	10000	0.6
470	0.99997	2.69	104826.19	104825.40	104822.89	220900	69.5
972	0.99997	13.82	455659.61	455656.38	455651.08	944784	650.5
1300	0.9997	279.55	819146.96	819143.51	-	15000	inf

$E(N)$ - результат работы алгоритма, E_{min} - минимально известное значение из литературы [**Iterative_procedure_table_s**]

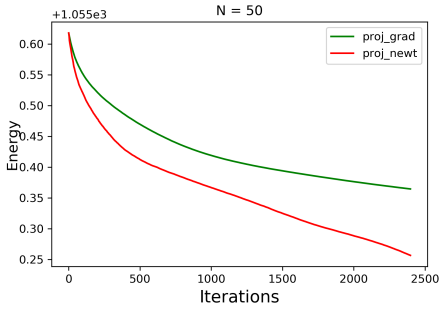

Рис. 12: Сравнение методов с градиентным и Ньютоновским шагами

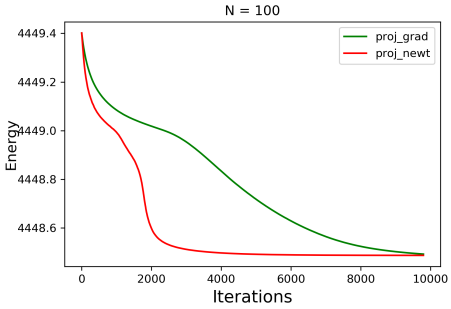

Рис. 13: Сравнение методов с градиентным и Ньютоновским шагами

В разделе 4 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 "количество итераций"$\leq N^2$. На Рис. 12, 13 изображены графики сходимости для двух методов при $s = 1$. Оказывается, что время работы алгоритмов 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 итераций, которое требуется методу $proj\_newt$, чтобы достичь значения функции, которое достиг $proj\_grad$ за N^2 итераций меньше чем $N^2/2$.

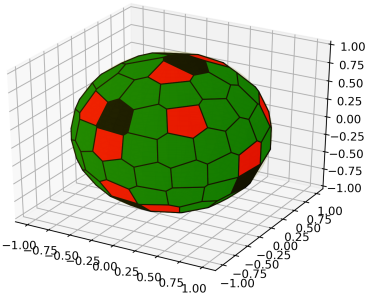

Рис. 14: Диаграмма Вороного до оптимизации, N = 100

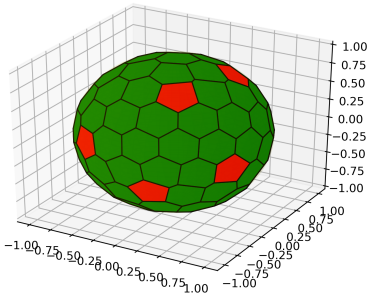

Рис. 15: Диаграмма Вороного после оптимизации, N = 100

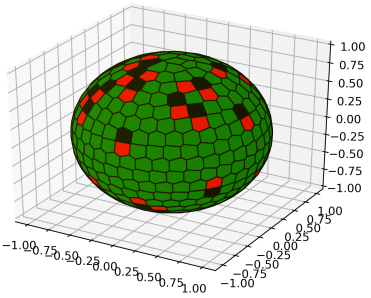

Рис. 16: Диаграмма Вороного до оптимизации, N = 470

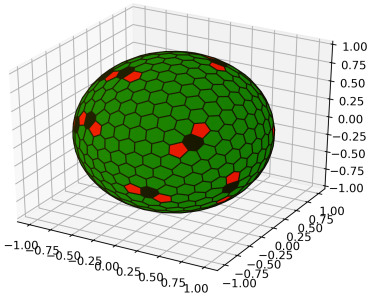

Рис. 17: Диаграмма Вороного после оптимизации, N = 470

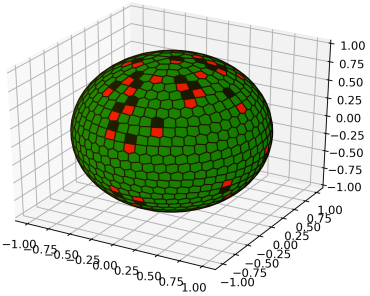

Рис. 18: Диаграмма Вороного до оптимизации, N = 972

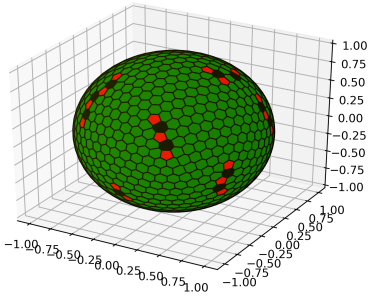

Рис. 19: Диаграмма Вороного после оптимизации, N = 972

N	Известный минимум	Минимум (метод 1)	Минимум (метод 2)	Итераций (1)	Итераций (2)	Время (1), сек.	Время (2), сек.
2	0.500000000	0.500000000	0.500000000	47	9	0.01	0.01
3	1.732050808	1.732050808	1.732050808	68	25	0.01	0.04
4	3.674234614	3.674234614	3.674234614	74	33	0.02	0.08
5	6.474691495	6.474691495	6.474691495	302	62	0.08	0.15
6	9.985281374	9.985281374	9.985281374	236	76	0.07	0.15
7	14.452977414	14.452977419	14.452977415	6923	1412	2.04	4.37
8	19.675287861	19.675287862	19.675287861	1175	363	0.39	1.10
9	25.759986531	25.759986532	25.759986531	1873	612	0.72	1.78
10	32.716949460	32.716949461	32.716949460	2095	1054	0.83	3.59
11	40.596450510	40.596450509	40.596450508	2656	1182	1.22	4.02
12	49.165253058	49.165253058	49.165253058	531	301	0.30	0.90
13	58.853230612	58.853230613	58.853230613	5968	4259	3.40	15.63
14	69.306363297	69.306363297	69.306363297	1562	1089	0.91	3.82
15	80.670244114	80.670244115	80.670244115	3075	1646	1.79	6.15
16	92.911655302	92.920353963	92.920353963	1921	1447	1.35	6.47
17	106.050404829	106.050404829	106.050404829	4956	3220	3.36	12.84
18	120.084467447	120.084467448	120.084467448	3618	2982	2.55	13.75
19	135.089467557	135.089467598	135.089467573	54150	37475	45.73	204.79
20	150.881568334	150.881568335	150.881568334	7888	4513	6.86	20.46
21	167.641622399	167.641622406	167.641622403	21011	16268	19.04	107.98
22	185.287536149	185.287536150	185.287536150	5039	3584	4.37	19.55
23	203.930190663	203.930190663	203.930190663	4318	2776	4.17	13.97
24	223.347074052	223.347074053	223.347074052	5792	3766	5.92	20.42
25	243.812760299	243.812760325	243.812760313	77672	48035	87.69	293.62
26	265.133326317	265.133326322	265.133326320	35677	22081	40.84	164.26
27	287.302615033	287.302615034	287.302615033	11121	6634	12.82	36.23
28	310.491542358	310.491542359	310.491542359	7290	4886	8.82	27.51
29	334.634439920	334.634439925	334.634439923	33329	21972	41.91	150.04
30	359.603945904	359.603945906	359.603945907	21705	16211	27.34	96.28
31	385.530838063	385.530838064	385.530838064	8830	3958	11.71	24.84
32	412.261274651	412.261274651	412.261274651	4638	2632	6.17	17.88
33	440.204057448	440.204057473	440.204057463	121994	80835	179.11	534.87
34	468.904853281	468.904853283	468.904853282	21336	11498	30.21	74.17
35	498.569872491	498.573454067	498.573454052	111480	62252	165.10	494.20
36	529.122408375	529.122408762	529.122408502	194400	173565	303.93	1529.87
37	560.618887731	560.618887735	560.618887733	27730	15097	42.05	121.43
38	593.038503566	593.038503567	593.038503567	10556	5695	16.55	43.95
39	626.389009017	626.389009018	626.389009017	10746	5670	17.35	39.49
40	660.675278835	660.675278835	660.675278835	14221	7569	23.16	53.30
41	695.916744342	695.916744342	695.916744342	11007	5798	18.61	43.47
42	732.078107544	732.078107544	732.078107544	24540	13251	42.10	86.15
43	769.190846459	769.190846479	769.190846468	161849	76440	293.69	756.65
44	807.174263085	807.174263096	807.174263090	85212	30703	153.64	295.41
45	846.188401061	846.188401076	846.188401069	266076	68438	577.57	744.37
46	886.167113639	886.170216025	886.171432426	31869	21742	62.98	184.46
47	927.059270680	927.072224603	927.072224580	58621	13125	119.45	109.35
48	968.713455344	968.713455346	968.713455345	26747	13547	55.07	122.35
49	1011.557182654	1011.557182655	1011.557182654	104883	28546	226.57	210.51
50	1055.182314726	1055.182314728	1055.182314727	29411	14368	63.83	121.55