

Глобальные градиентные методы оптимизации



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Факультет компьютерных наук
Прикладная математика и информатика
Москва

Судаков Илья Александрович
Февраль 2023

1 Постановка задачи

Многие методы оптимизации (координатный спуск, градиентный спуск, метод сопряженного градиента и др.) используют так называемый локальный поиск, т.е. поиск минимума одномерной функции вдоль луча в n -мерном евклидовом пространстве. При этом всегда находится локальный минимум. В проекте предлагается исследовать потенциал применения методов глобальной одномерной оптимизации для организации линейного поиска. В каких случаях удастся найти глобальный минимум многомерной функции, используя глобальный линейный поиск в контексте классических градиентных методов.

1. Программная реализация методов глобального линейного поиска. 2. Автоматическая проекция функции на направление с использованием пакета `sympy`. 3. Реализация алгоритма оптимизации многомерной функции. 4. Экспериментальное и теоретическое исследование.

"1. Алгоритм поиска глобального минимума одномерной функции. 2. Алгоритм оптимизации многомерной функции, основанный на комбинировании классического градиентного подхода и глобального линейного поиска. 3. Программная реализация алгоритма на языке Python. 4. Результаты вычислительных экспериментов. 5. Возможно написание статьи по результатам проекта."

2 Градиентный спуск

Пусть целевая функция имеет вид: $F(X) : X \rightarrow R$

Задача оптимизации задана в следующем виде: $F(X) \rightarrow \min$

Основная идея метода заключается в том, чтобы идти в направлении наискорейшего спуска, а это направление задаётся антиградиентом $-\nabla F$

и $X^{[i+1]} = X^{[i]} - \lambda^{[i]} \nabla F(X^{[i]})$ где $\lambda^{[j]}$ задаёт скорость градиентного спуска и может быть выбрана как

- постоянная, тогда метод может не сходиться
- убывать по какому-то закону
- гарантировать наискорейший спуск

2.1 Алгоритм градиентного спуска

- Задают начальное приближение и точность расчёта: X_0, ε
- Рассчитывают $X^{[i+1]} = X^{[i]} - \lambda^{[i]} \nabla F(X^{[i]})$
- Проверяют условие остановки (на усмотрение):
 - $|x^{[j]} - x^{[j+1]}| < \varepsilon$
 - $|F(x^{[j]}) - F(x^{[j+1]})| < \varepsilon$
 - $|\nabla F(X^{[i]})| < \varepsilon$
 - иначе переходят к пункту 2

Этот алгоритм будет реализован на python

2.2 Метод наискорейшего спуска

В случае наискорейшего спуска $\lambda^{[j]}$ определяется как:

$$\lambda^{[j]} = \operatorname{argmin} F(X^{[j+1]}) = \operatorname{argmin} F(X^{[j]} - \lambda \nabla F(X^{[j]}))$$

чтобы вычислить $\lambda^{[j]}$ будем использовать метод золотого сечения

3 Метод золотого сечения

3.1 Описание

Метод золотого сечения — это эффективная реализации троичного поиска, служащего для нахождения минимума/максимума унимодальной функции на отрезке. Лучше тем, что на каждой итерации вычисляется значение в одной, а не двух точках.

Пусть хотим найти минимум на отрезке $[a, b]$, тогда разобьем его на 3 части 2 точками s_1, s_2 , так чтобы при отсекании одного из крайних подотрезков, одна из точек осталась границей подотрезков, а соотношение длин подотрезков оставалось прежним.

$$|[a, s_1]| = |[s_2, b]| = l_1$$

$$|[s_1, s_2]| = l_2$$

$$\text{тогда: } \frac{l_2}{l_1} = \frac{l_1 - l_2}{l_2} \Rightarrow l_1^2 - l_1 \cdot l_2 - l_2^2 = 0 \Rightarrow \frac{l_1}{l_2} = \frac{1 + \sqrt{5}}{2} = \phi$$

3.2 Алгоритм тернарного поиска

написать ли

3.3 Оценка сложности

Так как на каждой итерации длина рассматриваемого отрезка умножается на $\frac{l_1 + l_2}{l_1 + l_2 + l_2} = \frac{3 + \sqrt{5}}{4 + 2\sqrt{5}} = \phi^{-1}$, то для достижения точности δ потребуется $\frac{\ln(\frac{|[a, b]|}{\delta})}{\ln(\phi)} \approx 2 \ln(\frac{|[a, b]|}{\delta})$ итераций.

3.4 Применение

А что если функция не унимодальная, тогда к чему сойдется этот метод? К точке, которая является одним из локальных минимумов функции либо одному из концов, в котором производная неположительная.

4 Глобальная одномерная оптимизация

Алгоритм поиска глобального минимума функции вдоль одного направления будет основываться на интервальном анализе. Перечислим основные определения и теоремы, которые нам пригодятся для построения алгоритма.

4.1 Определение

Интервалом $[a, b]$ называется следующее множество:

$$[a, b] := \{x \in \mathbb{R} | a \leq x \leq b\}$$

4.2 Арифметические свойства интервалов

- $x + y = \underline{x} + \underline{y}, \bar{x} + \bar{y}$
- $x - y = \underline{x} - \bar{y}, \bar{x} - \underline{y}$
- $x \cdot y = \min(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}), \max(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y})$
- $x \div y = x \cdot [1/\bar{y}, 1/\underline{y}]$, если $0 \notin y$

а как быть с другими функциями

4.3 Основная теорема интервальной арифметики

Пусть $f(x_1, \dots, x_n)$ - рациональная функция вещественных аргументов x_1, \dots, x_n , и для нее определен результат $F(\mathbf{X}_1, \dots, \mathbf{X}_n)$ подстановки вместо аргументов интервалов их изменения $(X_1, \dots, X_n) \subset \mathbb{R}^n$ и для (X_1, \dots, X_n) операции выполняются по правилам интервальной арифметики. Тогда

$$\{f\{x_1, \dots, x_n\} | x_1 \in \mathbf{X}_1, \dots, x_n \in \mathbf{X}_n\} \subset F(\mathbf{X}_1, \dots, \mathbf{X}_n)$$

как лучше преобразовать выражение чтобы функция быстрее сходилась

4.4 Утверждение

Пусть $f(x_1, \dots, x_n)$ - рациональная функция вещественных аргументов x_1, \dots, x_n , и $F(\mathbf{X}_1, \dots, \mathbf{X}_n)$ соответствующая ей интервальная функция, тогда выполняется монотонность по включению. Пусть $\mathbf{X}_1, \dots, \mathbf{X}_n$ и $\mathbf{Y}_1, \dots, \mathbf{Y}_n$, такие что $\mathbf{X}_1 \subset \mathbf{Y}_1, \dots, \mathbf{X}_n \subset \mathbf{Y}_n$, тогда

$$f\{\mathbf{X}_1, \dots, \mathbf{X}_n\} \subset f\{\mathbf{Y}_1, \dots, \mathbf{Y}_n\}$$

4.5 Определение

Пусть $N > 0$ натуральное число. Тогда если $\langle S_0, \dots, S_{n-1} \rangle$ семейство непустых подмножеств множества S , тогда будем называть его разбиением множества S . В частности, объединение S_0, \dots, S_{n-1} равно S , тогда последовательность $\langle S_0, \dots, S_{n-1} \rangle$ является покрытием S .

4.6 Теорема

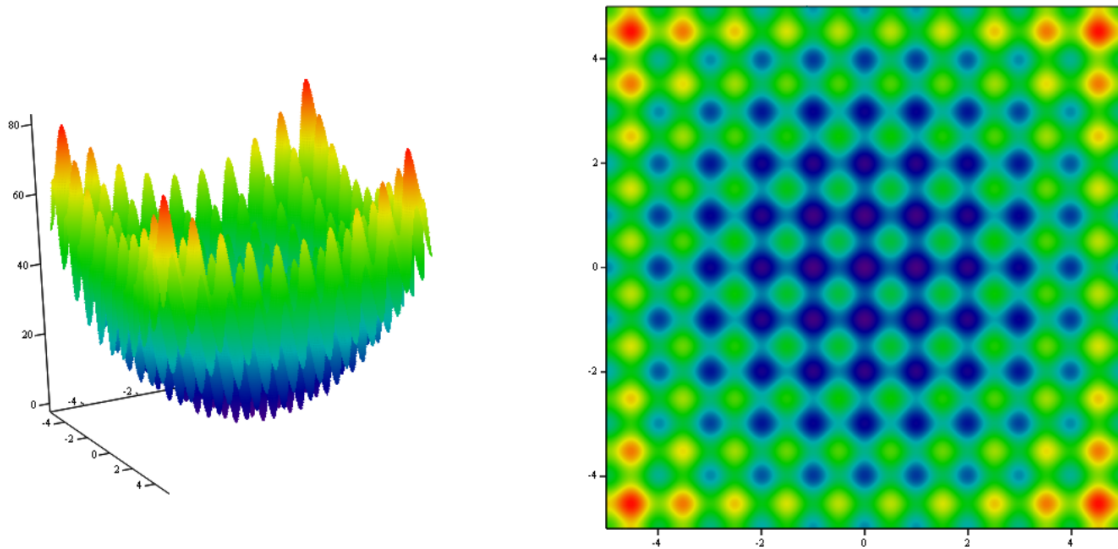
Рассмотрим задачу безусловной глобальной оптимизации. Пусть $\langle B_0, \dots, B_{n-1} \rangle$ семейство множеств, содержащее глобальный минимум, такое что упорядочено по возрастанию нижней границы $f(B_i)$ для $i = 0, 1, 2, \dots, N - 1$. Пусть U наименьшая из верхних значений функции для подмножеств $\langle f(B_0), \dots, f(B_{n-1}) \rangle$. Тогда интервал $[lb(f(B_0)), U]$ содержит глобальный минимум μ .

используя это напишем алгоритм поиска глобального минимума вдоль направления

5 Тестирование

5.1 Функция Растригина

5.1.1 График функции при $n = 2$



5.1.2 Формула

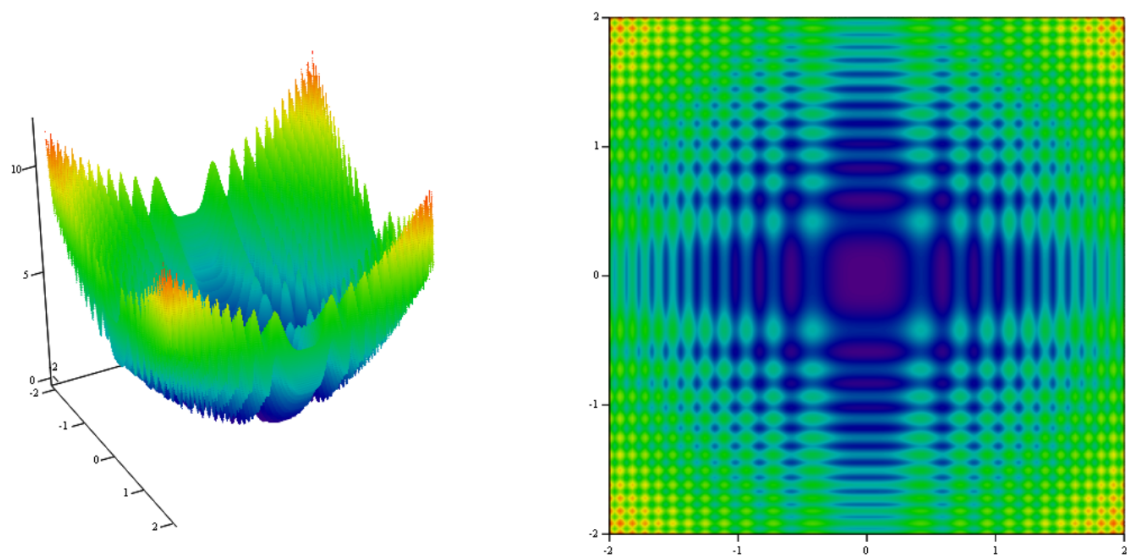
$$f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cdot \cos(2\pi \cdot x_i))$$

5.1.3 Результаты тестирования

п	точность	метод	время	минимум
5	10^{-3}	Golden ratio	2.22 ms	4.974821275706137
	10^{-3}	Moore-Skelboe	178 ms	9.238348694395881e-05
10	10^{-3}	Golden ratio	4.78 ms	9.94964255141227
	10^{-3}	Moore-Skelboe	736 ms	0.00018476697390212848
20	10^{-3}	Golden ratio	13.3 ms	19.899285102824635
	10^{-3}	Moore-Skelboe	2.52 s	0.0003695339478184678
30	10^{-3}	Golden ratio	24.1 ms	29.84892765423696
	10^{-3}	Moore-Skelboe	5.38 s	0.0005543009217348072
40	10^{-3}	Golden ratio	43.2 ms	39.79857020564907
	10^{-3}	Moore-Skelboe	9.58 s	0.0007390678956511465
50	10^{-3}	Golden ratio	58.8 ms	49.748212757061154
	10^{-3}	Moore-Skelboe	14.9 s	0.0009238348695674858

5.2 Функция Растригина новгородская

5.2.1 График функции при $n = 2$



5.2.2 Формула

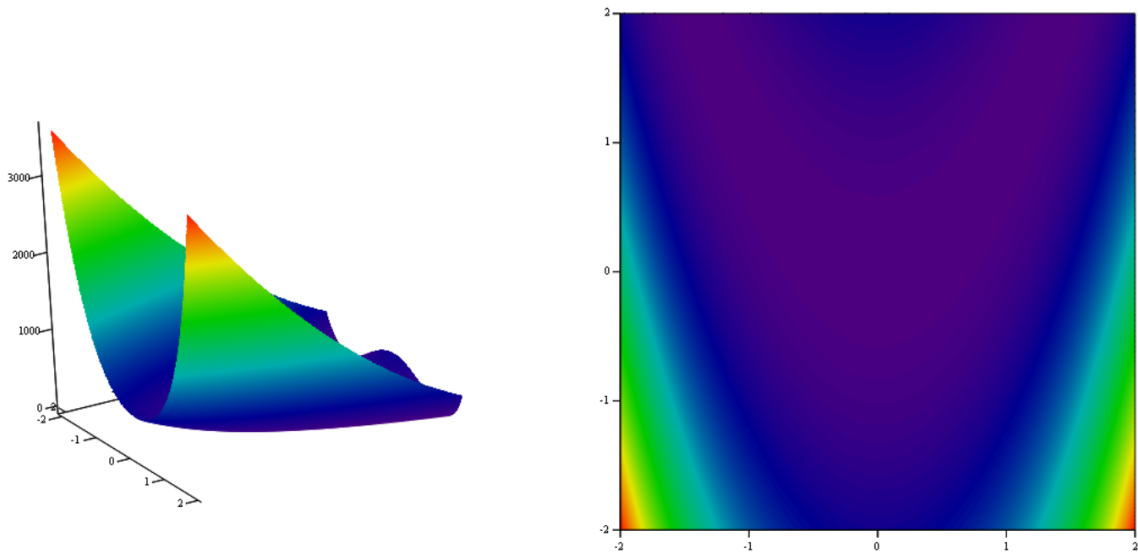
$$f(x) = n + \sum_{i=1}^n (x_i^2 - \cos(18 \cdot x_i^2))$$

5.2.3 Результаты тестирования

п	точность	метод	время	значение
5	10 ⁻³	Golden ratio	1.36 ms	1.542458003137213
	10 ⁻³	Moore-Skelboe	151 ms	0.0007449817996270092
10	10 ⁻³	Golden ratio	5.18 ms	3.0849160062744287
	10 ⁻³	Moore-Skelboe	636 ms	0.0014899635992544624
20	10 ⁻³	Golden ratio	20.8 ms	6.16983201254885
	10 ⁻³	Moore-Skelboe	1.94 s	0.002979927198509369
30	10 ⁻³	Golden ratio	37 ms	9.254748018823253
	10 ⁻³	Moore-Skelboe	4.54 s	0.0044698907977642754
40	10 ⁻³	Golden ratio	54.2 ms	12.339664025097658
	10 ⁻³	Moore-Skelboe	7.79 s	0.005959854397019182
50	10 ⁻³	Golden ratio	79.7 ms	15.424580031372061
	10 ⁻³	Moore-Skelboe	11.6 s	0.0074498179962740885

5.3 Функция Розенброка

5.3.1 График функции при $n = 2$



5.3.2 Формула

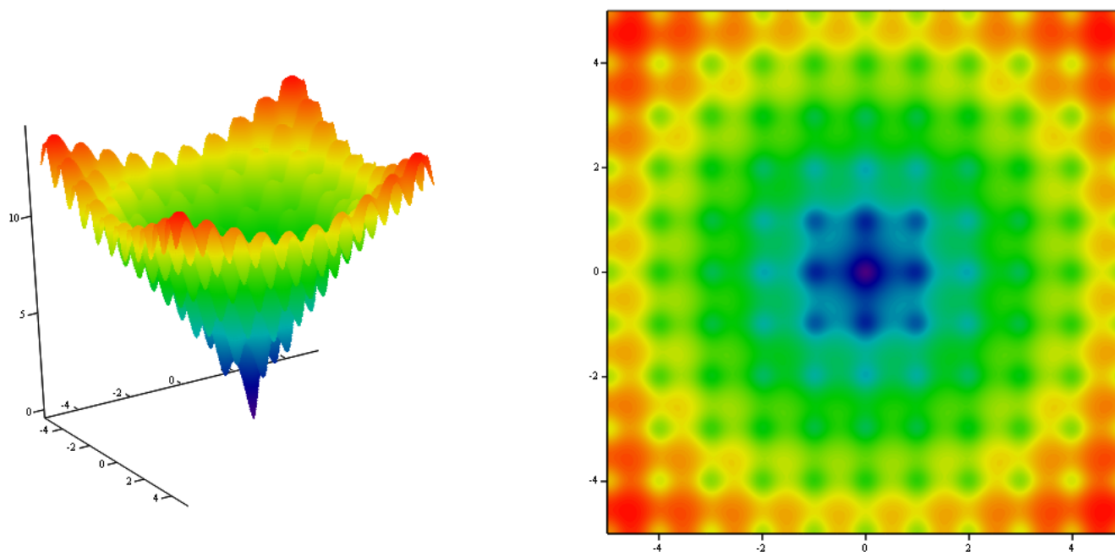
$$f(x) = \sum_{i=1}^{n-1} \left(100 (x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right)$$

5.3.3 Результаты тестирования

n	точность	метод	время	значение
2	10^{-3}	Golden ratio	1.12 ms	0.16956582889643249
	10^{-3}	Moore-Skelboe	155 ms	0.16036252999621464
3	10^{-3}	Golden ratio	5.95 ms	0.259397310396006
	10^{-3}	Moore-Skelboe	3.99 s	0.11254106932659813
4	10^{-3}	Golden ratio	17 ms	0.28064340055794645
	10^{-3}	Moore-Skelboe	8.75 s	0.11734786681147619
5	10^{-3}	Golden ratio	27.3 ms	0.2830939134564683
	10^{-3}	Moore-Skelboe	15.6 s	0.11851270446807871
6	10^{-3}	Golden ratio	40.4 ms	0.29047947563212634
	10^{-3}	Moore-Skelboe	24.5 s	0.11879389607067276
7	10^{-3}	Golden ratio	49.2 ms	0.29064024190256904
	10^{-3}	Moore-Skelboe	35.6 s	0.1136410373197746
8	10^{-3}	Golden ratio	18.5 ms	0.33244082803407593
	10^{-3}	Moore-Skelboe	41.7 s	0.14015453688727936

5.4 Функция Экли

5.4.1 График функции при $n = 2$



5.4.2 Формула

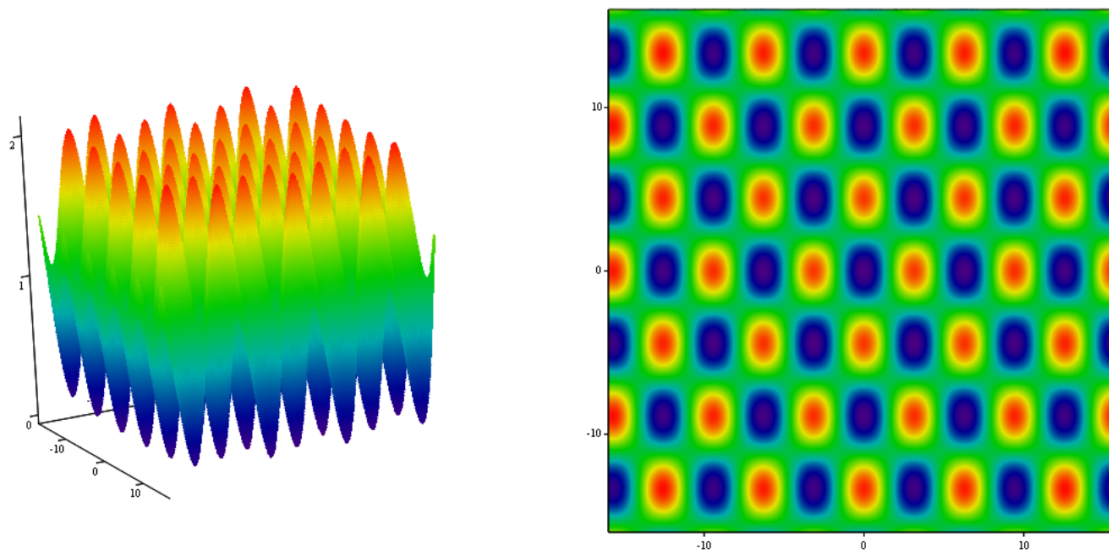
$$f(x) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^n \cos(2\pi \cdot x_i)}$$

5.4.3 Результаты тестирования

n	точность	метод	время	значение
5	10^{-3}	Golden ratio	2.41 ms	-4.440892098500626e-16
	10^{-3}	Moore-Skelboe	553 ms	0.0012256630393632229
20	10^{-3}	Golden ratio	9.06 ms	3.5744545080266543
	10^{-3}	Moore-Skelboe	1.35 s	0.0012256630393632229
20	10^{-3}	Golden ratio	30.6 ms	3.5744545080266543
	10^{-3}	Moore-Skelboe	3.41 s	0.0012256630393632229
30	10^{-3}	Golden ratio	58.5 ms	3.574454508026655
	10^{-3}	Moore-Skelboe	6.23 s	0.0012256630393645551
40	10^{-3}	Golden ratio	84.1 ms	3.574454508026654
	10^{-3}	Moore-Skelboe	10.6 s	0.0012256630393632229
50	10^{-3}	Golden ratio	124 ms	3.5744545080266525
	10^{-3}	Moore-Skelboe	14.9 s	0.0012256630393618906

5.5 Функция Гриванка

5.5.1 График функции при $n = 2$



5.5.2 Формула

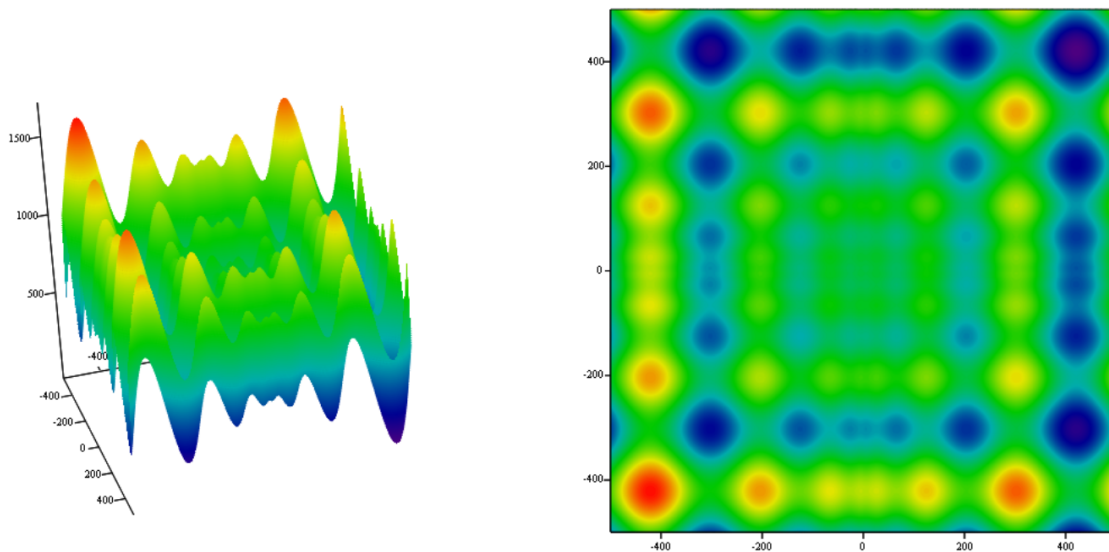
$$f(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

5.5.3 Результаты тестирования

п	точность	метод	время	значение
5	10 ⁻³	Golden ratio	2.41 ms	0.009864698515380743
	10 ⁻³	Moore-Skelboe	183 ms	1.0644240466817223e-07
10	10 ⁻³	Golden ratio	6.05 ms	0.009864698515380743
	10 ⁻³	Moore-Skelboe	739 ms	1.3662353537391425e-07
20	10 ⁻³	Golden ratio	21.4 ms	0.009864698515380743
	10 ⁻³	Moore-Skelboe	2.44 s	1.6799845647952338e-07
30	10 ⁻³	Golden ratio	40.2 ms	0.009864698515380743
	10 ⁻³	Moore-Skelboe	5.16 s	1.867295605917363e-07
40	10 ⁻³	Golden ratio	59.3 ms	0.009864698515380743
	10 ⁻³	Moore-Skelboe	9.09 s	2.0016648982768004e-07
50	10 ⁻³	Golden ratio	84.4 ms	0.009864698515380743
	10 ⁻³	Moore-Skelboe	14 s	2.1067470723501458e-07

5.6 Функция Швепеля

5.6.1 График функции при $n = 2$



5.6.2 Формула

$$f(x) = 418.9829n - \sum_{i=1}^n \left(x_i \sin \left(\sqrt{|x_i|} \right) \right)$$

5.6.3 Результаты тестирования

п	точность	метод	время	значение
2	10 ⁻³	Golden ratio	721 μs	236.8766946858181
	10 ⁻³	Moore-Skelboe	4.98 s	2.5455285594944144e-05
3	10 ⁻³	Golden ratio	800 μs	355.3150420287272
	10 ⁻³	Moore-Skelboe	9.41 s	3.8182928392416216e-05
4	10 ⁻³	Golden ratio	1.44 ms	473.7533893716363
	10 ⁻³	Moore-Skelboe	15 s	5.091057118988829e-05
5	10 ⁻³	Golden ratio	1.4 ms	592.1917367145454
	10 ⁻³	Moore-Skelboe	22 s	6.363821398736036e-05
6	10 ⁻³	Golden ratio	2.19 ms	710.6300840574543
	10 ⁻³	Moore-Skelboe	31.1 s	7.636585655745876e-05
7	10 ⁻³	Golden ratio	1.88 ms	829.0684314003631
	10 ⁻³	Moore-Skelboe	40 s	8.909349912755715e-05
8	10 ⁻³	Golden ratio	4.92 ms	947.5067787432718
	10 ⁻³	Moore-Skelboe	52 s	0.00010182114169765555

Список литературы

- Сергиенко А. Б. Тестовые функции для глобальной оптимизации.
- Пантелеев, Летова: Методы оптимизации в примерах и задачах.
- Ratschek, Rokne. New Computer Methods for Global Optimization.