Министерство образования и науки Российской Федерации
федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
«Сибирский государственный аэрокосмический университет
имени академика М.Ф. Решетнева»

Сергиенко Антон Борисович

Тестовые функции для глобальной оптимизации. v.1.29

Красноярск – 2014

Оглавление

Условные обозначения

$a \in A$ — элемент a принадлежит множеству A.

$\bar{x}$ — обозначение вектора.

$\arg f(x)$ — возвращает аргумент x, при котором функция принимает значение $f(x)$.

$Random(X)$ — случайный выбор элемента из множества X с равной вероятностью.

$Random\left(\{x^i \mid p^i\}\right)$ — случайный выбор элемента x^i из множества X, при условии, что каждый элемент $x^i \in X$ имеет вероятность выбора равную p^i, то есть это обозначение равнозначно предыдущему.

$random(a, b)$ — случайное действительное число из интервала $[a; b]$.

$int(a)$ — целая часть действительного числа a.

$\mu(X)$ — мощность множества X.

Замечание. Оператор присваивания обозначается через знак «=», так же как и знак равенства.

Замечание. Индексация всех массивов в документе начинается с 1. Это стоит помнить при реализации алгоритма на С-подобных языках программирования, где индексация начинается с нуля.

Замечание. Вызывание трех функций: $Random(X)$, $Random\left(\{x_i \mid p_i\}\right)$, $random(a, b)$ – происходит каждый раз, когда по ходу выполнения формул, они встречаются. Если формула итерационная, то нельзя перед ее вызовом один раз определить, например, $random(a, b)$ как константу и потом её использовать на протяжении всех итераций неизменной.

Замечание. Надстрочный индекс может обозначать как возведение в степень, так и индекс элемента. Конкретное обозначение определяется в контексте текста, в котором используется формула с надстрочным индексом.

Замечание. Если у нас имеется множество векторов, то подстрочный индекс обозначает номер компоненты конкретного вектора, а надстрочный индекс обозначает номер вектора во множестве, например, $\bar{x}^i \in X$ $(i = \overline{1, N})$, $\bar{x}^i_j \in \{0; 1\}$, $(j = \overline{1, n})$. В случае, если вектор имеет свое обозначение в виде подстрочной надписи, то компоненты вектора проставляются за скобками, например, $(\bar{x}_{max})_j = 0$ $(j = \overline{1, n})$.

Замечание. При выводе матриц и векторов элементы могут разделяться как пробелом, так и точкой с запятой, то есть обе записи $\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}^{\mathrm{T}}$ и $(1; 1; 1; 1; 1; 1; 1; 1)^{\mathrm{T}}$ допустимы.

Замечание. При выводе множеств элементы разделяются только точкой с запятой, то есть допустима только такая запись: $\{1; 1; 1; 1; 1; 1; 1; 1\}^{\mathrm{T}}$.

Введение

В данном документе рассмотрено множество тестовых функций, которые можно использовать для проведения исследований алгоритмов оптимизации. К каждой функции дано подробное описание, график (если это возможно), свойств и параметров, которые позволят единообразно проводить сравнения разных алгоритмов оптимизации во избежания несостыковок с точки зрения разного понимания нахождения ошибки, точности работы алгоритмом.

Данный документ представляет его версию **1.29** от 4 января 2014 г.

Последнюю версию документа можно найти по адресу:

https://github.com/Harrix/HarrixTestFunctions

Там же можно найти реализацию тестовых функция в среде Mathcad.

Тестовые функции реализованы на языке C++ в библиотеке **HarrixMathLibrary** в разделе «Тестовые функции для оптимизации», которую можно найти по адресу:

https://github.com/Harrix/HarrixMathLibrary.

Все библиографические материалы, которые используются в документе, приведены в виде скриншотов, скринов, документов в папке **_Biblio** на https://github.com/Harrix/HarrixTestFunctions.

С автором можно связаться по адресу sergienkoanton@mail.ru или http://vk.com/harrix.

Сайт автора, где публикуются последние новости: http://blog.harrix.org/, а проекты располагаются по адресу http://harrix.org/.

Глава 1

Список функций

1.1 Список по идентификатору

Вначале идут функции многомерной оптимизации, потом двумерной, далее одномерной. В конце идут функции бинарной оптимизации.

1. **MHL_TestFunction_Ackley** — функция Ackley;
2. **MHL_TestFunction_GaussianQuartic** — функция Gaussian quartic;
3. **MHL_TestFunction_Griewangk** — функция Griwанка;
4. **MHL_TestFunction_HyperEllipsoid** — гипер-эллипсоид;
5. **MHL_TestFunction_ParaboloidOfRevolution** — эллиптический параболоид;
6. **MHL_TestFunction_Rastrigin** — функция Растригина;
7. **MHL_TestFunction_RastriginNovgorod** — функция Растригина новгородская;
8. **MHL_TestFunction_Rosenbrock** — функция Розенброка;
9. **MHL_TestFunction_RotatedHyperEllipsoid** — развернутый гипер-эллипсоид;
10. **MHL_TestFunction_Schwefel** — функция Швефеля;
11. **MHL_TestFunction_StepFunction** — функция Step (модифицированная версия De Jong 3);
12. **MHL_TestFunction_AdditivePotential** — аддитивная потенциальная функция;
13. **MHL_TestFunction_EggHolder** — функция Egg Holder;
14. **MHL_TestFunction_Himmelblau** — функция Химмельблау;
15. **MHL_TestFunction_InvertedRosenbrock** — перевернутая функция Розенброка;
16. **MHL_TestFunction_Katnikov** — функция Катникова;
17. **MHL_TestFunction_Multiextremal3** — функция Multiextremal3;
18. **MHL_TestFunction_Multiextremal4** — функция Multiextremal4;

19. **MHL_TestFunction_MultiplicativePotential** — мультипликативная потенциальная функция;
20. **MHL_TestFunction_Rana** — функция Rana;
21. **MHL_TestFunction_RastriginWithChange** — функция Растригина с изменением коэффициентов;
22. **MHL_TestFunction_RastriginWithTurning** — функция Растригина овражная с поворотом осей;
23. **MHL_TestFunction_ReverseGriewank** — функция ReverseGriewank;
24. **MHL_TestFunction_ShekelsFoxholes** — функция "Лисьи норы"Шекеля;
25. **MHL_TestFunction_Sombrero** — функция Сомбреро;
26. **MHL_TestFunction_Multiextremal** — функция Multiextremal;
27. **MHL_TestFunction_Multiextremal2** — функция Multiextremal2;
28. **MHL_TestFunction_Wave** — волна;
29. **MHL_TestFunction_SumVector** — сумма всех элементов бинарного вектора.

1.2 Список формул

1. **MHL_TestFunction_Ackley** — функция Ackley:

$$f\left(\bar{x}\right) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}\bar{x}_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}cos(2\pi\cdot\bar{x}_i)}. \tag{1.1}$$

2. **MHL_TestFunction_GaussianQuartic** — функция Gaussian quartic:

$$f\left(\bar{x}\right) = \sum_{i=1}^{n}\left(i\cdot\bar{x}_i^4\right) + rnorm\left(0,1\right). \tag{1.2}$$

3. **MHL_TestFunction_Griewangk** — функция Griванка:

$$f\left(\bar{x}\right) = \sum_{i=1}^{n}\frac{\bar{x}_i^2}{4000} - \prod_{i=1}^{n}\cos\left(\frac{\bar{x}_i}{\sqrt{i}}\right) + 1. \tag{1.3}$$

4. **MHL_TestFunction_HyperEllipsoid** — гипер-эллипсоид:

$$f\left(\bar{x}\right) = \sum_{i=1}^{n}\left(i\cdot\bar{x}_i\right)^2. \tag{1.4}$$

5. **MHL_TestFunction_ParaboloidOfRevolution** — эллиптический параболоид:

$$f\left(\bar{x}\right) = \sum_{i=1}^{n}\bar{x}_i^2. \tag{1.5}$$

6. **MHL_TestFunction_Rastrigin** — функция Растригина:

$$f\left(\bar{x}\right)=10n+\sum_{i=1}^{n}\left(\bar{x}_i^2-10\cdot\cos\left(2\pi\cdot\bar{x}_i\right)\right). \tag{1.6}$$

7. **MHL_TestFunction_RastriginNovgorod** — функция Растригина новгородская:

$$f\left(\bar{x}\right)=n+\sum_{i=1}^{n}\left(\bar{x}_i^2-\cos\left(18\cdot\bar{x}_i^2\right)\right). \tag{1.7}$$

8. **MHL_TestFunction_Rosenbrock** — функция Розенброка:

$$f\left(\bar{x}\right)=\sum_{i=1}^{n-1}\left(100\left(\bar{x}_{i+1}-\bar{x}_i^2\right)^2+\left(1-\bar{x}_i\right)^2\right). \tag{1.8}$$

9. **MHL_TestFunction_RotatedHyperEllipsoid** — развернутый гипер-эллипсоид:

$$f\left(\bar{x}\right)=\sum_{i=1}^{n}\left(\sum_{j=1}^{j}\bar{x}_j\right)^2. \tag{1.9}$$

10. **MHL_TestFunction_Schwefel** — функция Швефеля:

$$f\left(\bar{x}\right)=418.9829n-\sum_{i=1}^{n}\left(\bar{x}_i\sin\left(\sqrt{\left|\bar{x}_i\right|}\right)\right). \tag{1.10}$$

11. **MHL_TestFunction_StepFunction** — функция Step (модифицированная версия De Jong 3):

$$f\left(\bar{x}\right)=\begin{cases}\sum_{i=1}^{n}\left(int\left(\bar{x}_i\right)\right)^2, \text{ если} \sum_{i=1}^{n}\left|int\left(\bar{x}_i\right)\right|\neq 0;\\ \left(\sum_{i=1}^{n}\left|\bar{x}_i\right|\right)-1, \text{ иначе.}\end{cases} \tag{1.11}$$

12. **MHL_TestFunction_AdditivePotential** — аддитивная потенциальная функция:

$$f\left(\bar{x}\right)=z\left(\bar{x}_1\right)+z\left(\bar{x}_2\right), \text{ где} \tag{1.12}$$

$$z\left(v\right)=-\frac{1}{\left(v-1\right)^2+0.2}-\frac{1}{2\left(v-2\right)^2+0.15}-\frac{1}{3\left(v-3\right)^2+0.3}.$$

13. **MHL_TestFunction_EggHolder** — функция Egg Holder:

$$f\left(\bar{x}\right)=-\bar{x}_1\sin\left(\sqrt{\left|\bar{x}_1-47-\bar{x}_2\right|}\right)-\left(\bar{x}_2+47\right)\sin\left(\sqrt{\left|\frac{\bar{x}_1}{2}+47+\bar{x}_2\right|}\right). \tag{1.13}$$

14. **MHL_TestFunction_Himmelblau** — функция Химмельблау:

$$f\left(\bar{x}\right)=\left(\bar{x}_1^2+\bar{x}_2-11\right)^2+\left(\bar{x}_1+\bar{x}^2-7\right)^2. \tag{1.14}$$

15. **MHL_TestFunction_InvertedRosenbrock** — перевернутая функция Розенброка:

$$f(\bar{x}) = \frac{-100}{100\left(\bar{x}_1^2 - \bar{x}_2\right) + \left(1. - \bar{x}_1\right)^2 + 600}. \quad (1.15)$$

16. **MHL_TestFunction_Katnikov** — функция Катникова:

$$f(\bar{x}) = 0.5\left(\bar{x}_1^2 + \bar{x}_2^2\right)\left(2A + A\cos\left(1.5\bar{x}_1\right)\cos\left(3.14\bar{x}_2\right) + A\cos\left(\sqrt{5}\bar{x}_1\right)\cos\left(3.5\bar{x}_2\right)\right). \quad (1.16)$$

17. **MHL_TestFunction_Multiextremal3** — функция Multiextremal3:

$$f(\bar{x}) = \bar{x}_1^2\left|\sin\left(2\bar{x}_1\right)\right| + \bar{x}_2^2\left|\sin\left(2\bar{x}_2\right)\right| - \frac{1}{5\bar{x}_1^2 + 5\bar{x}_2^2 + 0.2} + 5. \quad (1.17)$$

18. **MHL_TestFunction_Multiextremal4** — функция Multiextremal4:

$$\begin{aligned} f(\bar{x}) =& 0.5\left(\bar{x}_1^2 + \bar{x}_1\bar{x}_2 + \bar{x}_2^2\right)\left(1 + 0.5\cos\left(1.5\bar{x}_1\right)\cos\left(3.2\bar{x}_1\bar{x}_2\right)\cos\left(3.14\bar{x}_2\right) + \right. \\ & \left. + 0.5\cos\left(2.2\bar{x}_1\right)\cos\left(4.8\bar{x}_1\bar{x}_2\right)\cos\left(3.5\bar{x}_2\right)\right). \end{aligned} \quad (1.18)$$

19. **MHL_TestFunction_MultiplicativePotential** — мультипликативная потенциальная функция:

$$f(\bar{x}) = -z\left(\bar{x}_1\right)\cdot z\left(\bar{x}_2\right), \text{ где} \quad (1.19)$$

$$z(v) = -\frac{1}{(v-1)^2 + 0.2} - \frac{1}{2(v-2)^2 + 0.15} - \frac{1}{3(v-3)^2 + 0.3}.$$

20. **MHL_TestFunction_Rana** — функция Rana:

$$f(\bar{x}) = \bar{x}_1\sin\left(\sqrt{\left|\bar{x}_2 + 1 - \bar{x}_1\right|}\right)\cos\left(\sqrt{\left|\bar{x}_2 + 1 + \bar{x}_1\right|}\right) + \quad (1.20)$$

$$+ \left(\bar{x}_2 + 1\right)\cos\left(\sqrt{\left|\bar{x}_2 + 1 - \bar{x}_1\right|}\right)\sin\left(\sqrt{\left|\bar{x}_2 + 1 + x\right|}\right). \quad (1.21)$$

21. **MHL_TestFunction_RastriginWithChange** — функция Растригина с изменением коэффициентов:

$$f(\bar{x}) = 0.1\bar{x}_1^2 + 0.1\bar{x}_2^2 - 4\cos\left(0.8\bar{x}_1\right) - 4\cos\left(0.8\bar{x}_2\right) + 8. \quad (1.22)$$

22. **MHL_TestFunction_RastriginWithTurning** — функция Растригина овражная с поворотом осей:

$$\begin{aligned} f(\bar{x}) =& \left(0.1K_{\bar{x}_1}A\left(\bar{x}_1, \bar{x}_2\right)\right)^2 + \left(0.1K_{\bar{x}_2}B\left(\bar{x}_1, \bar{x}_2\right)\right)^2 - \\ & - 4\cos\left(0.8K_{\bar{x}_1}A\left(\bar{x}_1, \bar{x}_2\right)\right) - 4\cos\left(0.8K_{\bar{x}_2}B\left(\bar{x}_1, \bar{x}_2\right)\right) + 8, \text{ где} \\ & A\left(\bar{x}_1, \bar{x}_2\right) = \bar{x}_1\cos(\alpha) - \bar{x}_2\sin(\alpha), \\ & B\left(\bar{x}_1, \bar{x}_2\right) = \bar{x}_1\sin(\alpha) + \bar{x}_2\cos(\alpha). \end{aligned} \quad (1.23)$$

23. **MHL_TestFunction_ReverseGriewank** — функция ReverseGriewank:

$$f(\bar{x}) = \frac{1}{\frac{\bar{x}_1^2 + \bar{x}_2^2}{200} - cos\left(\bar{x}_0\right)cos\left(\frac{\bar{x}_2}{\sqrt{2}}\right) + 2}, \text{ где} \quad (1.24)$$

24. **MHL_TestFunction_ShekelsFoxholes** — функция "Лисьи норы"Шекеля:

$$f\left(\bar{x}\right) = \frac{1}{\frac{1}{K} + \sum_{j=1}^{25} \frac{1}{j + \left(\bar{x}_1 - A_{1,j}\right)^6 + \left(\bar{x}_2 - A_{2,j}\right)^6}}, \text{ где} \tag{1.25}$$

$$A = \left(\begin{smallmatrix} -32 & -16 & 0 & 16 & 32 & -32 & -16 & 0 & 16 & 32 & -32 & -16 & 0 & 16 & 32 & -32 & -16 & 0 & 16 & 32 & -32 & -16 & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & -16 & -16 & -16 & -16 & 0 & 0 & 0 & 0 & 0 & 16 & 16 & 16 & 16 & 16 & 32 & 32 & 32 & 32 & 32 \end{smallmatrix}\right).$$

25. **MHL_TestFunction_Sombrero** — функция Сомбреро:

$$f\left(\bar{x}\right) = \frac{1 - sin\left(\sqrt{\bar{x}_1^2 + \bar{x}_2^2}\right)^2}{1 + 0.001\left(\bar{x}_1^2 + \bar{x}_2^2\right)}. \tag{1.26}$$

26. **MHL_TestFunction_Multiextremal** — функция Multiextremal:

$$f\left(\bar{x}\right) = 0.05\left(x-1\right)^2 + \left(3 - 2.9e^{-2.77257x^2}\right)\left(1 - \cos\left(x\left(4 - 50e^{-2.77257x^2}\right)\right)\right). \tag{1.27}$$

27. **MHL_TestFunction_Multiextremal2** — функция Multiextremal2:

$$f\left(\bar{x}\right) = 1 - 0.5\cos\left(1.5\left(10x - 0.3\right)\right)\cos\left(31.4x\right) + 0.5\cos\left(\sqrt{5}\cdot 10x\right)\cos\left(35x\right). \tag{1.28}$$

28. **MHL_TestFunction_Wave** — волна:

$$f\left(\bar{x}\right) = e^{-\bar{x}_1^2} + 0.01\cos\left(200\cdot\bar{x}_1\right). \tag{1.29}$$

29. **MHL_TestFunction_SumVector** — сумма всех элементов бинарного вектора:

$$f\left(\bar{x}\right) = \sum_{i=1}^{n} \bar{x}_i. \tag{1.30}$$

Глава 2

Задачи вещественной оптимизации

2.1 Функция Ackley

2.1.1 Описание функции

Идентификатор: MHL_TestFunction_Ackley.

Наименование: Функция Ackley.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}\bar{x}_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}cos(2\pi\cdot\bar{x}_i)},\ \text{где} \tag{2.1}$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = -5$, $Right_j = 5$, $j = \overline{1,n}$.

Обозначение: $\bar{x}$ — вещественный вектор;

n — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min} = \arg\min_{\bar{x}\in X} f(\bar{x})$.

Точка минимума: $\bar{x}_{min} = (0, 0, \ldots, 0)^{\mathrm{T}}$, то есть $(\bar{x}_{min})_j = 0$ $(j = \overline{1,n})$.

Минимум функции: $f(\bar{x}_{min}) = 0$.

График: Рисунок 2.1 нас 18 стр.

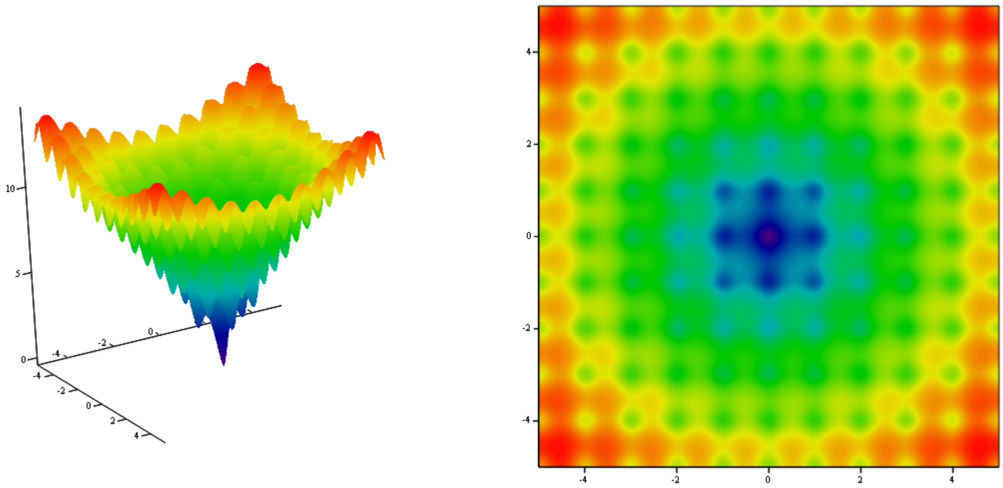


Рисунок 2.1. Функция Ackley

2.1.2 Параметры для алгоритмов оптимизации

Точность вычислений: $\varepsilon = 0.025$.

Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) : $NumberOfParts_j = 4095$ $(j = \overline{1, n})$.

Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) : $(k_2)_j = 12$ $(j = \overline{1, n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } (k_2)_j \in \mathbb{N}, \left(j = \overline{1, n}\right).$$

2.1.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

Подзадача №2: $n = 3$.

Подзадача №3: $n = 4$.

Подзадача №4: $n = 5$.

Подзадача №5: $n = 10$.

Подзадача №6: $n = 20$.

Подзадача №7: $n = 30$.

2.1.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно ($k = \overline{1, N}$). Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1, n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N} \left(\frac{\sqrt{\sum_{j=1}^{n} \left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n} \right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N} \left|f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.1.5 Свойства задачи

Условной или безусловной оптимизации:	Задача безусловной оптимизации.
Одномерной или многомерной оптимизации:	Многомерной: n.
Функция унимодальная или многоэкстремальная:	Функция многоэкстремальная.
Функция стохастическая или нет:	Функция не стохастическая.
Особенности:	Нет.

2.1.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.1. Код функции MHL_TestFunction_Ackley

```
double MHL_TestFunction_Ackley(double *x, int VMHL_N)
{
/*
```

```
Функция многих переменных: Ackley.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - указатель на исходный массив;
 VMHL_N - размер массива x.
Возвращаемое значение:
 Значение тестовой функции в точке x.
*/
double VMHL_Result;
double f1,f2=0;
f1=exp(-0.2*sqrt(TMHL_SumSquareVector(x,VMHL_N)/double(VMHL_N)));
for (int i=0;i<VMHL_N;i++) f2=f2+cos(2.*MHL_PI*x[i]);
f2=exp(f2/double(VMHL_N));
VMHL_Result=20.+exp(1)-20.*f1-f2;
return VMHL_Result;
}
```

2.1.7 Ссылки

Данная функция приводится в следующих источниках:

1. [1, стр. 5] — Empirical review of standard benchmark functions using evolutionary global optimization.
2. [2] — Ackley's Function.

2.2 Функция Gaussian quartic

2.2.1 Описание функции

Идентификатор: MHL_TestFunction_GaussianQuartic.

Наименование: Функция Gaussian quartic.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right)=\sum_{i=1}^{n}\left(i\cdot\bar{x}_i^4\right)+rnorm\left(0,1\right),\ \text{где} \tag{2.2}$$

$\bar{x}\in X$, $\bar{x}_j\in\left[Left_j;Right_j\right]$, $Left_j=-2$, $Right_j=2$, $j=\overline{1,n}$, $rnorm\left(0,1\right)$ — возвращает случайное число, распределенное по нормальному закону с параметрами $\left(0,1\right)$.

Обозначение: $\bar{x}$ — вещественный вектор;

n — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min} = \arg\min_{\bar{x} \in X} f\left(\bar{x}\right)$.

Точка минимума: $\bar{x}_{min} = \left(0, 0, \ldots, 0\right)^{\mathrm{T}}$, то есть $\left(\bar{x}_{min}\right)_j = 0$ $(j = \overline{1, n})$.

Минимум функции: $f\left(\bar{x}_{min}\right) = 0$.

График: Рисунок 2.2 нас 21 стр.

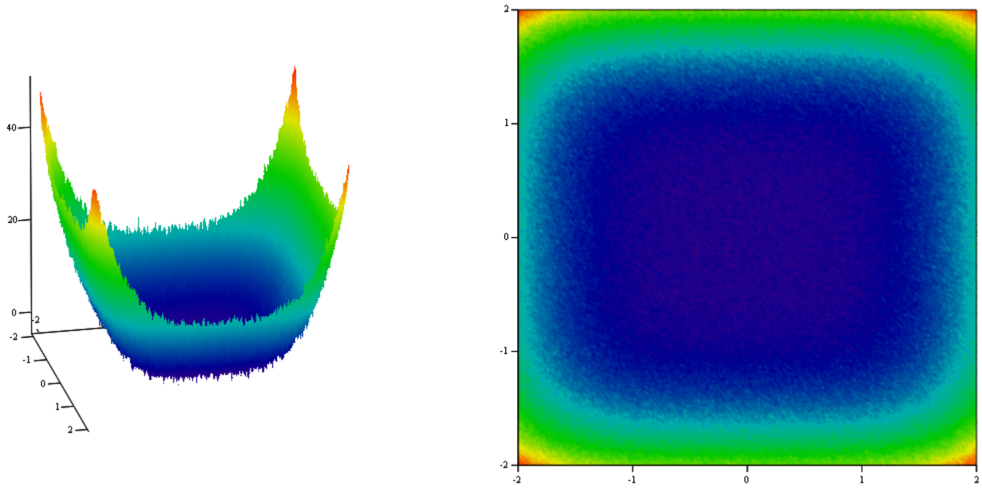


Рисунок 2.2. Функция Gaussian quartic

2.2.2 Параметры для алгоритмов оптимизации

Точность вычислений: $\varepsilon = 0.01$.

Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) : $NumberOfParts_j = 4095$ $(j = \overline{1, n})$.

Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) : $\left(k_2\right)_j = 12$ $(j = \overline{1, n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{\left(k_2\right)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } \left(k_2\right)_j \in \mathbb{N}, \left(j = \overline{1, n}\right).$$

2.2.3 Основная задача и подзадачи

Изменяемый параметр:	n — размерность вещественного вектора.
Значение в основной задаче:	$n = 2$.
Подзадача №2:	$n = 3$.
Подзадача №3:	$n = 4$.
Подзадача №4:	$n = 5$.
Подзадача №5:	$n = 10$.
Подзадача №6:	$n = 20$.
Подзадача №7:	$n = 30$.

2.2.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно ($k = \overline{1, N}$). Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1, n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N} \left(\frac{\sqrt{\sum_{j=1}^{n} \left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j \right)^2}}{n} \right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N} \left| f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.2.5 Свойства задачи

Условной или безусловной оптимизации:	Задача безусловной оптимизации.
Одномерной или многомерной оптимизации:	Многомерной: n.
Функция унимодальная или многоэкстремальная:	Функция унимодальная.
Функция стохастическая или нет:	Функция стохастическая.
Особенности:	Локальные минимумы очень похожи на глобальный по своему значению.

2.2.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.2. Код функции MHL_TestFunction_GaussianQuartic

```
double MHL_TestFunction_GaussianQuartic(double *x, int VMHL_N)
{
/*
Функция многих переменных: функция Gaussian quartic.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - указатель на исходный массив;
 VMHL_N - размер массива x.
Возвращаемое значение:
 Значение тестовой функции в точке x.
*/
double VMHL_Result=0;

for (int i=0;i<VMHL_N;i++) VMHL_Result+=(i+1)*x[i]*x[i]*x[i]*x[i];

VMHL_Result+=MHL_RandomNormal(0, 1);

return VMHL_Result;
}
```

2.2.7 Ссылки

Данная функция приводится в следующих источниках:

1. [3] — De Jong's test suite.
2. [4, стр. 25] — A Literature Survey of Benchmark Functions For Global Optimization Problems.

2.3 Функция Griewank

2.3.1 Описание функции

Идентификатор: MHL_TestFunction_Griewangk.

Наименование: Функция Griewank.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right)=\sum_{i=1}^{n}\frac{\bar{x}_i^2}{4000}-\prod_{i=1}^{n}\cos\left(\frac{\bar{x}_i}{\sqrt{i}}\right)+1,\ \text{где} \tag{2.3}$$

$\bar{x}\in X$, $\bar{x}_j\in\left[Left_j;Right_j\right]$, $Left_j=-16$, $Right_j=16$, $j=\overline{1,n}$.

Обозначение: $\bar{x}$ — вещественный вектор;

n — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min}=\arg\min_{\bar{x}\in X}f\left(\bar{x}\right)$.

Точка минимума: $\bar{x}_{min}=\left(0,0,\ldots,0\right)^{\mathrm{T}}$, то есть $\left(\bar{x}_{min}\right)_j=0$ $\left(j=\overline{1,n}\right)$.

Минимум функции: $f\left(\bar{x}_{min}\right)=0$.

График: Рисунок 2.3 на с 24 стр.

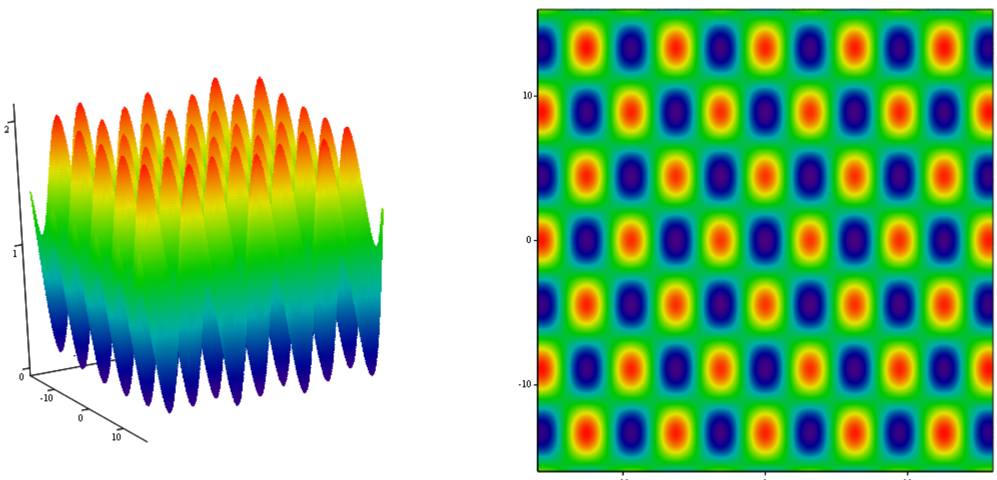


Рисунок 2.3. Функция Griewank

2.3.2 Параметры для алгоритмов оптимизации

Точность вычислений:	$\varepsilon = 0.08.$
Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) :	$NumberOfParts_j = 4095$ $(j = \overline{1,n})$.
Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) :	$(k_2)_j = 12$ $(j = \overline{1,n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } \left(k_2\right)_j \in \mathbb{N}, \left(j = \overline{1,n}\right).$$

2.3.3 Основная задача и подзадачи

Изменяемый параметр:	n — размерность вещественного вектора.
Значение в основной задаче:	$n = 2.$
Подзадача №2:	$n = 3.$
Подзадача №3:	$n = 4.$
Подзадача №4:	$n = 5.$
Подзадача №5:	$n = 10.$
Подзадача №6:	$n = 20.$
Подзадача №7:	$n = 30.$

2.3.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно $(k = \overline{1,N})$. Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1,n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N} \left(\frac{\sqrt{\sum_{j=1}^{n} \left(\left(\bar{x}_{submin}^k \right)_j - \left(\bar{x}_{min} \right)_j \right)^2}}{n} \right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N} \left| f\left(\bar{x}_{submin}^k \right) - f\left(\bar{x}_{min} \right) \right|}{N}.$$

2.3.5 Свойства задачи

Условной или безусловной оптимизации:	Задача безусловной оптимизации.
Одномерной или многомерной оптимизации:	Многомерной: n.
Функция унимодальная или многоэкстремальная:	Функция многоэкстремальная.
Функция стохастическая или нет:	Функция не стохастическая.
Особенности:	Локальные минимумы очень похожи на глобальный по своему значению.

2.3.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.3. Код функции MHL_TestFunction_Griewangk

```
double MHL_TestFunction_Griewangk(double *x, int VMHL_N)
{
/*
Функция многих переменных: функция Гриванка.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - указатель на исходный массив;
 VMHL_N - размер массива x.
Возвращаемое значение:
 Значение тестовой функции в точке x.
*/
double VMHL_Result=0;
double f=1;
int i;

for (i=0;i<VMHL_N;i++) VMHL_Result+=x[i]*x[i];
VMHL_Result/=4000.;
```

```
for (i=0;i<VMHL_N;i++) f=f*cos(x[i]/sqrt(i+1));

VMHL_Result=VMHL_Result-f+1.;

return VMHL_Result;
}
```

2.3.7 Ссылки

Данная функция приводится в следующих источниках:

1. [5, стр. 7] — GEATbx Examples. Examples of Objective Functions.
2. [4, стр. 16] — A Literature Survey of Benchmark Functions For Global Optimization Problems.

2.4 Функция Гипер-эллипсоид

2.4.1 Описание функции

Идентификатор: MHL_TestFunction_HyperEllipsoid.

Наименование: Гипер-эллипсоид.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = \sum_{i=1}^{n} (i \cdot \bar{x}_i)^2, \text{ где} \tag{2.4}$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = -5$, $Right_j = 5$, $j = \overline{1, n}$.

Обозначение: $\bar{x}$ — вещественный вектор;

n — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min} = \arg\min_{\bar{x} \in X} f(\bar{x})$.

Точка минимума: $\bar{x}_{min} = (0, 0, \ldots, 0)^{\mathrm{T}}$, то есть $(\bar{x}_{min})_j = 0$ $(j = \overline{1, n})$.

Минимум функции: $f(\bar{x}_{min}) = 0$.

График: Рисунок 2.4 нас 28 стр.

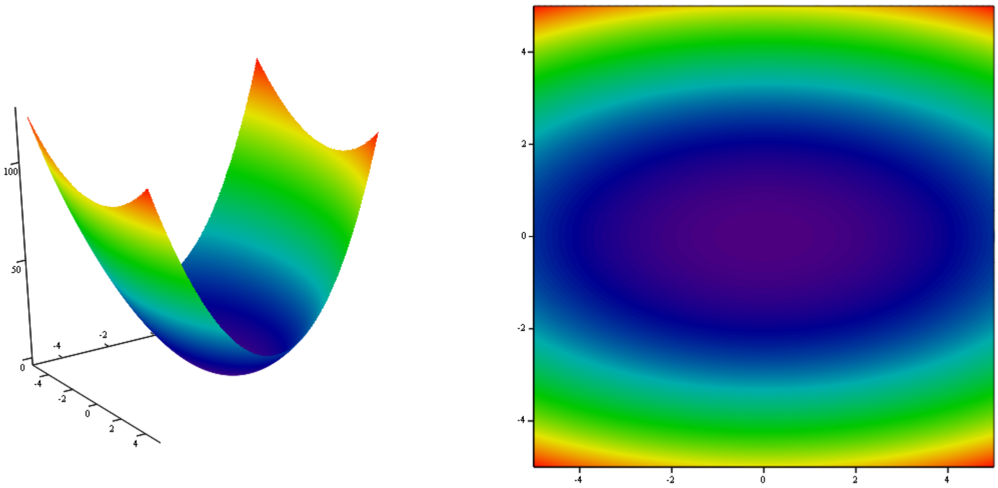


Рисунок 2.4. Гипер-эллипсоид

2.4.2 Параметры для алгоритмов оптимизации

Точность вычислений: $\varepsilon = 0.025$.

Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) : $NumberOfParts_j = 4095$ $(j = \overline{1,n})$.

Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) : $(k_2)_j = 12$ $(j = \overline{1,n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } \left(k_2\right)_j \in \mathbb{N}, \left(j = \overline{1,n}\right).$$

2.4.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

Подзадача №2: $n = 3$.

Подзадача №3: $n = 4$.

Подзадача №4: $n = 5$.

Подзадача №5: $n = 10$.

Подзадача №6: $n = 20$.

Подзадача №7: $n = 30$.

2.4.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно $(k = \overline{1,N})$. Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1,n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N}\left(\frac{\sqrt{\sum_{j=1}^{n}\left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n}\right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N}\left|f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.4.5 Свойства задачи

Условной или безусловной оптимизации:	Задача безусловной оптимизации.
Одномерной или многомерной оптимизации:	Многомерной: n.
Функция унимодальная или многоэкстремальная:	Функция унимодальная.
Функция стохастическая или нет:	Функция не стохастическая.
Особенности:	Нет.

2.4.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.4. Код функции MHL_TestFunction_HyperEllipsoid

```
double MHL_TestFunction_HyperEllipsoid(double *x, int VMHL_N)
{
/*
```

```
Функция многих переменных: Гипер-эллипсоид.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - указатель на исходный массив;
 VMHL_N - размер массива x.
Возвращаемое значение:
 Значение тестовой функции в точке x.
*/
double VMHL_Result=0;

for (int i=0;i<VMHL_N;i++)
   VMHL_Result += (i+1)*(i+1)*x[i]*x[i];

return VMHL_Result;
}
```

2.4.7 Ссылки

В данном виде тестовую функцию в литературе не нашел. Обычно используется несколько иной вид этой функции, когда i не возводится в квадрат.

2.5 Эллиптический параболоид

2.5.1 Описание функции

Идентификатор: MHL_TestFunction_ParaboloidOfRevolution.

Наименование: Эллиптический параболоид.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right) = \sum_{i=1}^{n} \bar{x}_i^2, \text{ где} \tag{2.5}$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = -2$, $Right_j = 2$, $j = \overline{1, n}$.

Обозначение: $\bar{x}$ — вещественный вектор;

n — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min} = \arg\min_{\bar{x} \in X} f\left(\bar{x}\right)$.

Точка минимума: $\bar{x}_{min} = (0, 0, \ldots, 0)^{\mathrm{T}}$, то есть $\left(\bar{x}_{min}\right)_j = 0$ $(j = \overline{1, n})$.

Минимум функции: $f\left(\bar{x}_{min}\right) = 0$.

График: Рисунок 2.5 нас 31 стр.

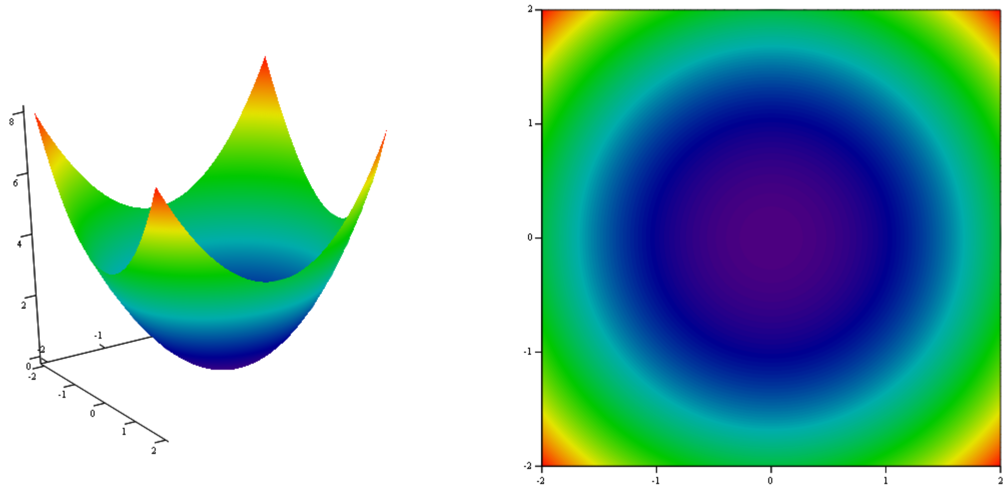


Рисунок 2.5. Эллиптический параболоид

2.5.2 Параметры для алгоритмов оптимизации

Точность вычислений:	$\varepsilon = 0.01$.
Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) :	$NumberOfParts_j = 4095$ $(j = \overline{1,n})$.
Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) :	$(k_2)_j = 12$ $(j = \overline{1,n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\,(Right_j - Left_j)}{\varepsilon}, \text{где } (k_2)_j \in \mathbb{N}, \left(j = \overline{1,n}\right).$$

2.5.3 Основная задача и подзадачи

Изменяемый параметр:	n — размерность вещественного вектора.
Значение в основной задаче:	$n = 2$.
Подзадача №2:	$n = 3$.
Подзадача №3:	$n = 4$.
Подзадача №4:	$n = 5$.
Подзадача №5:	$n = 10$.
Подзадача №6:	$n = 20$.
Подзадача №7:	$n = 30$.

2.5.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно ($k = \overline{1, N}$). Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1, n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N}\left(\frac{\sqrt{\sum_{j=1}^{n}\left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n}\right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N}\left|f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.5.5 Свойства задачи

Условной или безусловной оптимизации:	Задача безусловной оптимизации.
Одномерной или многомерной оптимизации:	Многомерной: n.
Функция унимодальная или многоэкстремальная:	Функция унимодальная.
Функция стохастическая или нет:	Функция не стохастическая.
Особенности:	Нет.

2.5.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.5. Код функции MHL_TestFunction_ParaboloidOfRevolution

```
double MHL_TestFunction_ParaboloidOfRevolution(double *x, int VMHL_N)
{
/*
```

```
Функция многих переменных: Эллиптический параболоид.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - указатель на исходный массив;
 VMHL_N - размер массива x.
Возвращаемое значение:
 Значение тестовой функции в точке x.
*/
double VMHL_Result=0;
for (int i=0;i<VMHL_N;i++) VMHL_Result+=x[i]*x[i];
return VMHL_Result;
}
```

2.5.7 Ссылки

Данная функция приводится в следующих источниках:

1. [6] — Paraboloid.

2.6 Функция Растригина

2.6.1 Описание функции

Идентификатор: MHL_TestFunction_Rastrigin.

Наименование: Функция Растригина.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right)=10n+\sum_{i=1}^{n}\left(\bar{x}_i^2-10\cdot\cos\left(2\pi\cdot\bar{x}_i\right)\right),\ \text{где} \tag{2.6}$$

$\bar{x}\in X$, $\bar{x}_j\in[Left_j;Right_j]$, $Left_j=-5$, $Right_j=5$, $j=\overline{1,n}$.

Обозначение: $\bar{x}$ — вещественный вектор;

n — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min}=\arg\min_{\bar{x}\in X}f\left(\bar{x}\right)$.

Точка минимума: $\bar{x}_{min}=\left(0,0,\ldots,0\right)^{\mathrm{T}}$, то есть $\left(\bar{x}_{min}\right)_j=0$ $(j=\overline{1,n})$.

Минимум функции: $f\left(\bar{x}_{min}\right)=0$.

График: Рисунок 2.6 нас 34 стр.

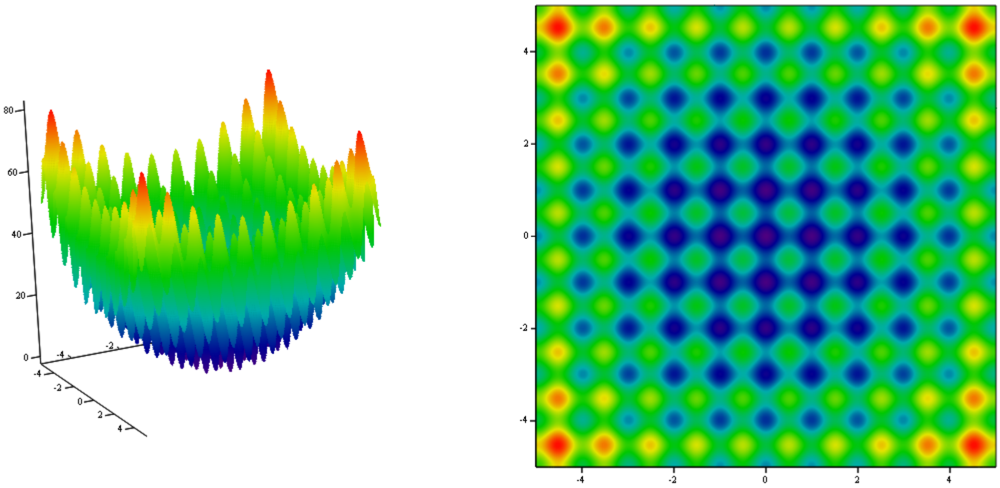


Рисунок 2.6. Функция Растригина

2.6.2 Параметры для алгоритмов оптимизации

Точность вычислений: $\varepsilon = 0.025$.

Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) : $NumberOfParts_j = 4095$ $(j = \overline{1,n})$.

Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) : $(k_2)_j = 12$ $(j = \overline{1,n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } \left(k_2\right)_j \in \mathbb{N}, \left(j = \overline{1,n}\right).$$

2.6.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

Подзадача №2: $n = 3$.

Подзадача №3: $n = 4$.

Подзадача №4: $n = 5$.

Подзадача №5: $n = 10$.

Подзадача №6: $n = 20$.

Подзадача №7: $n = 30$.

2.6.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно ($k = \overline{1, N}$). Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1, n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N} \left(\frac{\sqrt{\sum_{j=1}^{n} \left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j \right)^2}}{n} \right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N} \left| f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right) \right|}{N}.$$

2.6.5 Свойства задачи

Условной или безусловной оптимизации:	Задача безусловной оптимизации.
Одномерной или многомерной оптимизации:	Многомерной: n.
Функция унимодальная или многоэкстремальная:	Функция многоэкстремальная.
Функция стохастическая или нет:	Функция не стохастическая.
Особенности:	Нет.

2.6.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.6. Код функции MHL_TestFunction_Rastrigin

```
double MHL_TestFunction_Rastrigin(double *x, int VMHL_N)
{
/*
```

```
Функция многих переменных: функция Растригина.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - указатель на исходный массив;
 VMHL_N - размер массива x.
Возвращаемое значение:
 Значение тестовой функции в точке x.
*/
double VMHL_Result=0;
for (int i=0;i<VMHL_N;i++) VMHL_Result+=x[i]*x[i]-10.*cos(2.*MHL_PI*x[i]);
VMHL_Result+=10*VMHL_N;
return VMHL_Result;
}
```

2.6.7 Ссылки

Данная функция приводится в следующих источниках:

1. [7] — Rastrigin function.
2. [8] — Non-linear Continuous Multi-Extremal Optimization.
3. [9] — Parametric Optimization.

2.7 Функция Растригина новгородская

2.7.1 Описание функции

Идентификатор: MHL_TestFunction_RastriginNovgorod.

Наименование: Функция Растригина новгородская.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right) = n + \sum_{i=1}^{n} \left(\bar{x}_i^2 - \cos\left(18 \cdot \bar{x}_i^2\right)\right), \text{ где} \tag{2.7}$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = -2$, $Right_j = 2$, $j = \overline{1, n}$.

Обозначение: $\bar{x}$ — вещественный вектор;

n — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min} = \arg\min_{\bar{x} \in X} f\left(\bar{x}\right)$.

Точка минимума: $\bar{x}_{min} = \left(0, 0, \ldots, 0\right)^{\mathrm{T}}$, то есть $\left(\bar{x}_{min}\right)_j = 0$ $(j = \overline{1, n})$.

Минимум функции: $f\left(\bar{x}_{min}\right) = 0$.

График: Рисунок 2.7 нас 37 стр.

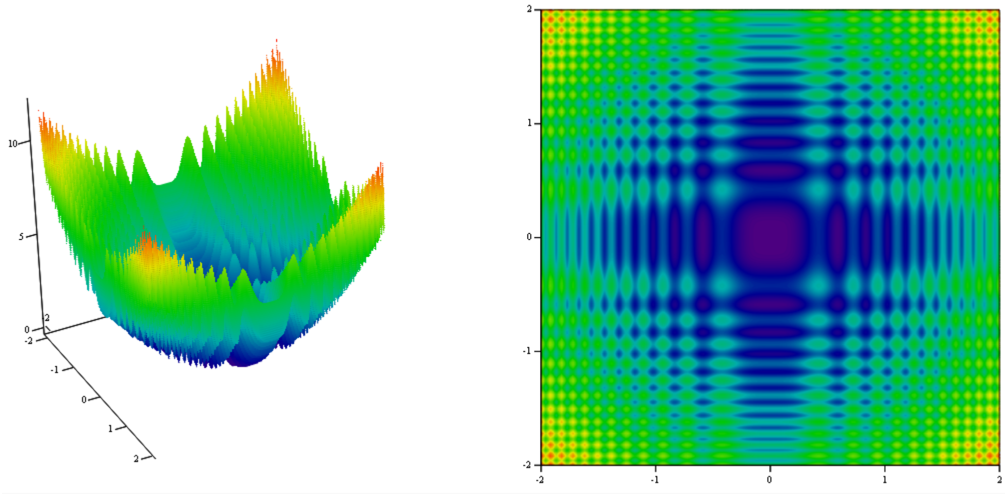


Рисунок 2.7. Функция Растригина новгородская

2.7.2 Параметры для алгоритмов оптимизации

Точность вычислений: $\varepsilon = 0.01$.

Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) : $NumberOfParts_j = 4095$ $(j = \overline{1,n})$.

Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) : $(k_2)_j = 12$ $(j = \overline{1,n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } (k_2)_j \in \mathbb{N}, \left(j = \overline{1,n}\right).$$

2.7.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

Подзадача №2: $n = 3$.

Подзадача №3: $n = 4$.

Подзадача №4: $n = 5$.

Подзадача №5: $n = 10$.

Подзадача №6: $n = 20$.

Подзадача №7: $n = 30$.

2.7.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно ($k = \overline{1,N}$). Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1,n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N}\left(\frac{\sqrt{\sum_{j=1}^{n}\left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n}\right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N}\left|f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.7.5 Свойства задачи

Условной или безусловной оптимизации:	Задача безусловной оптимизации.
Одномерной или многомерной оптимизации:	Многомерной: n.
Функция унимодальная или многоэкстремальная:	Функция многоэкстремальная.
Функция стохастическая или нет:	Функция не стохастическая.
Особенности:	Нет.

2.7.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.7. Код функции MHL_TestFunction_RastriginNovgorod

```
double MHL_TestFunction_RastriginNovgorod(double *x, int VMHL_N)
{
/*
```

```
Функция многих переменных: функция Растригина новгородская.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - указатель на исходный массив;
 VMHL_N - размер массива x.
Возвращаемое значение:
 Значение тестовой функции в точке x.
*/
double VMHL_Result=0;

for (int i=0;i<VMHL_N;i++)
    VMHL_Result+=x[i]*x[i]-cos(18.*x[i]*x[i]);

VMHL_Result+=VMHL_N;

return VMHL_Result;
}
```

2.7.7 Ссылки

Данная функция проскочила в переписке с Василием Рябовым (varjag84@mail.ru) из Нижнего Новгорода.

2.8 Функция Розенброка

2.8.1 Описание функции

Идентификатор: MHL_TestFunction_Rosenbrock.

Наименование: Функция Розенброка.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right)=\sum_{i=1}^{n-1}\left(100\left(\bar{x}_{i+1}-\bar{x}_i^2\right)^2+\left(1-\bar{x}_i\right)^2\right),\ \text{где} \tag{2.8}$$

$\bar{x}\in X$, $\bar{x}_j\in[Left_j;Right_j]$, $Left_j=-2$, $Right_j=2$, $j=\overline{1,n}$.

Обозначение: $\bar{x}$ — вещественный вектор;

n — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min}=\arg\min_{\bar{x}\in X} f\left(\bar{x}\right)$.

Точка минимума: $\bar{x}_{min}=\left(1,1,\ldots,1\right)^{\mathrm{T}}$, то есть $\left(\bar{x}_{min}\right)_j=1$ $(j=\overline{1,n})$.

Минимум функции: $f\left(\bar{x}_{min}\right)=0$.

График: Рисунок 2.8 на с 40 стр.

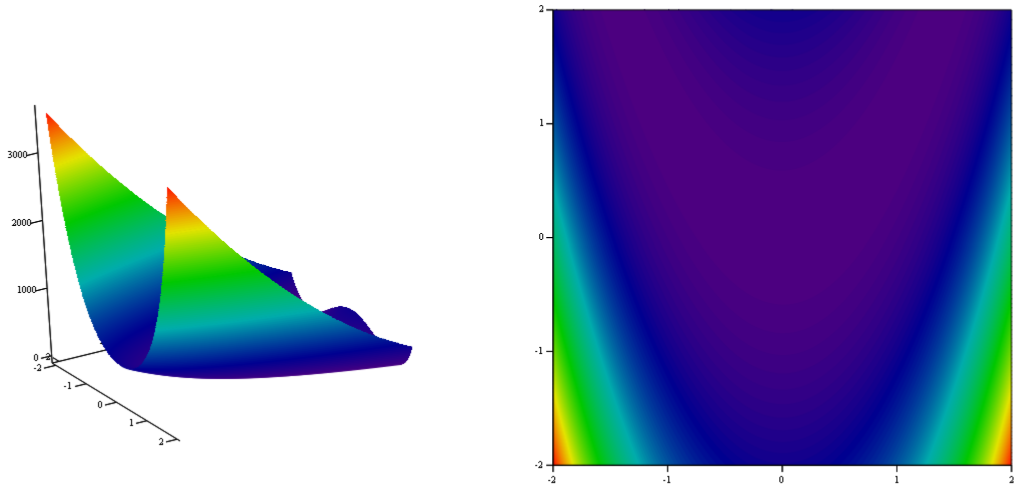


Рисунок 2.8. Функция Розенброка

2.8.2 Параметры для алгоритмов оптимизации

Точность вычислений: $\varepsilon = 0.01$.

Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) : $NumberOfParts_j = 4095$ $(j = \overline{1, n})$.

Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) : $(k_2)_j = 12$ $(j = \overline{1, n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\,(Right_j - Left_j)}{\varepsilon}, \text{где } (k_2)_j \in \mathbb{N}, \left(j = \overline{1, n}\right).$$

2.8.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

Подзадача №2: $n = 3$.

Подзадача №3: $n = 4$.

Подзадача №4: $n = 5$.

Подзадача №5: $n = 10$.

Подзадача №6: $n = 20$.

Подзадача №7: $n = 30$.

2.8.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно ($k=\overline{1,N}$). Используем три вида ошибок:

Надёжность:

$$R=\frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N},\text{ где}$$

$$S\left(\bar{x}^k_{submin}\right)=\begin{cases}1,\text{ если }\left|\left(\bar{x}^k_{submin}\right)_j-\left(\bar{x}_{min}\right)_j\right|<\varepsilon, j=\overline{1,n};\\0,\text{ иначе.}\end{cases}$$

Ошибка по входным параметрам:

$$E_x=\frac{\sum_{k=1}^{N}\left(\frac{\sqrt{\sum_{j=1}^{n}\left(\left(\bar{x}^k_{submin}\right)_j-\left(\bar{x}_{min}\right)_j\right)^2}}{n}\right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f=\frac{\sum_{k=1}^{N}\left|f\left(\bar{x}^k_{submin}\right)-f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.8.5 Свойства задачи

Условной или безусловной оптимизации:	Задача безусловной оптимизации.
Одномерной или многомерной оптимизации:	Многомерной: n.
Функция унимодальная или многоэкстремальная:	Функция многоэкстремальная.
Функция стохастическая или нет:	Функция не стохастическая.
Особенности:	Нет.

2.8.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.8. Код функции MHL_TestFunction_Rosenbrock

```
double MHL_TestFunction_Rosenbrock(double *x, int VMHL_N)
{
/*
```

```
Функция многих переменных: функция Розенброка.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - указатель на исходный массив;
 VMHL_N - размер массива x.
Возвращаемое значение:
 Значение тестовой функции в точке x.
*/
double VMHL_Result=0;
for (int i=0;i<VMHL_N-1;i++) VMHL_Result+=100.*(x[i+1]-x[i]*x[i])*(x[i+1]-x[i]*x[i])
   +(1.-x[i])*(1.-x[i]);
return VMHL_Result;
}
```

2.8.7 Ссылки

Данная функция приводится в следующих источниках:

1. [10] — Rosenbrock function.
2. [11] — Rosenbrock Function.

2.9 Развернутый гипер-эллипсоид

2.9.1 Описание функции

Идентификатор:	MHL_TestFunction_RotatedHyperEllipsoid.
Наименование:	Развернутый гипер-эллипсоид.
Тип:	Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right)=\sum_{i=1}^{n}\left(\sum_{j=1}^{j}\bar{x}_j\right)^2,\text{ где} \tag{2.9}$$

$\bar{x}\in X$, $\bar{x}_j\in[Left_j;Right_j]$, $Left_j=-5$, $Right_j=5$, $j=\overline{1,n}$.

Обозначение:	$\bar{x}$ — вещественный вектор;
	n — размерность вещественного вектора.
Решаемая задача оптимизации:	$\bar{x}_{min}=\arg\min_{\bar{x}\in X} f\left(\bar{x}\right)$.
Точка минимума:	$\bar{x}_{min}=\left(0,0,\ldots,0\right)^{\mathrm{T}}$, то есть $\left(\bar{x}_{min}\right)_j=0$ $(j=\overline{1,n})$.
Минимум функции:	$f\left(\bar{x}_{min}\right)=0$.
График:	Рисунок 2.9 нас 43 стр.

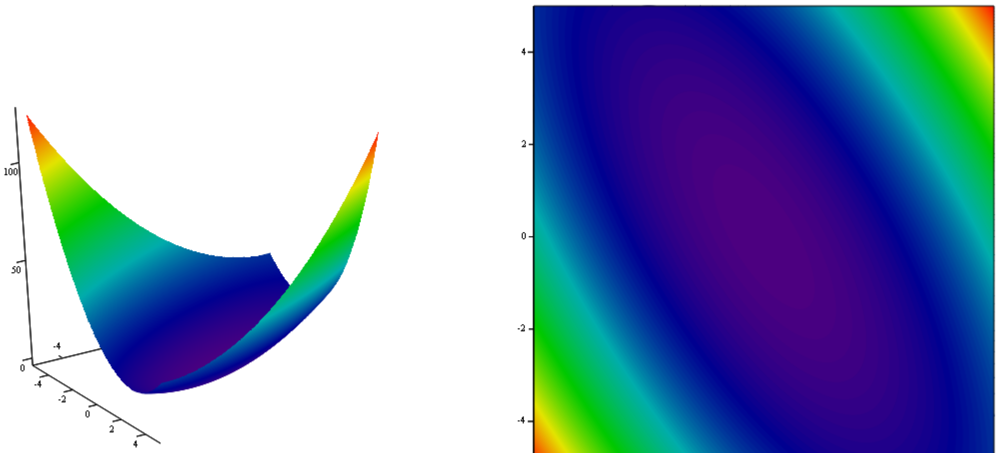


Рисунок 2.9. Развернутый гипер-эллипсоид

2.9.2 Параметры для алгоритмов оптимизации

Точность вычислений:	$\varepsilon = 0.025$.
Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) :	$NumberOfParts_j = 4095 \ (j = \overline{1, n})$.
Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) :	$(k_2)_j = 12 \ (j = \overline{1, n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } (k_2)_j \in \mathbb{N}, \left(j = \overline{1, n}\right).$$

2.9.3 Основная задача и подзадачи

Изменяемый параметр:	n — размерность вещественного вектора.
Значение в основной задаче:	$n = 2$.
Подзадача №2:	$n = 3$.
Подзадача №3:	$n = 4$.
Подзадача №4:	$n = 5$.
Подзадача №5:	$n = 10$.
Подзадача №6:	$n = 20$.
Подзадача №7:	$n = 30$.

2.9.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно ($k = \overline{1, N}$). Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1, n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N} \left(\frac{\sqrt{\sum_{j=1}^{n} \left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j \right)^2}}{n} \right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N} \left| f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right) \right|}{N}.$$

2.9.5 Свойства задачи

Условной или безусловной оптимизации:	Задача безусловной оптимизации.
Одномерной или многомерной оптимизации:	Многомерной: n.
Функция унимодальная или многоэкстремальная:	Функция унимодальная.
Функция стохастическая или нет:	Функция не стохастическая.
Особенности:	Нет.

2.9.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.9. Код функции MHL_TestFunction_RotatedHyperEllipsoid

```
double MHL_TestFunction_RotatedHyperEllipsoid(double *x, int VMHL_N)
{
/*
```

```
Функция многих переменных: Развернутый гипер-эллипсоид.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - указатель на исходный массив;
 VMHL_N - размер массива x.
Возвращаемое значение:
 Значение тестовой функции в точке x.
*/
double VMHL_Result=0;
double f;

for (int i=0;i<VMHL_N;i++)
 {
 f=0;
 for (int j=0;j<i+1;j++)
    f += x[j];
 VMHL_Result += f*f;
 }

return VMHL_Result;
}
```

2.9.7 Ссылки

Данная функция приводится в следующих источниках:

1. [5, стр. 4] — GEATbx Examples. Examples of Objective Functions.

Обратите внимание, что иногда под названием Rotated Hyper Ellipsoid встречается (например, [12]) неправильно записанная функция в виде:

$$f(\bar{x}) = \sum_{i=1}^{n} \sum_{j=1}^{j} \bar{x}_j^2, \text{ где}$$

Данная функция не является развернутой по своему внешнему виду, поэтому автор склонен считать эту реализацию ошибочной.

2.10 Функция Швефеля

2.10.1 Описание функции

Идентификатор:	MHL_TestFunction_Schwefel.
Наименование:	Функция Швефеля.
Тип:	Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right) = 418.9829n - \sum_{i=1}^{n}\left(\bar{x}_i \sin\left(\sqrt{|\bar{x}_i|}\right)\right), \text{ где} \tag{2.10}$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = -500$, $Right_j = 500$, $j = \overline{1,n}$.

Обозначение: $\bar{x}$ — вещественный вектор;

n — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min} = \arg\min_{\bar{x}\in X} f\left(\bar{x}\right)$.

Точка минимума: $\bar{x}_{min} = (420.968746, 420.968746, \ldots, 420.968746)^{\mathrm{T}}$, то есть $\left(\bar{x}_{min}\right)_j = 420.968746$ $(j = \overline{1,n})$.

Минимум функции: $f\left(\bar{x}_{min}\right) = 0.0000255$, если $n = 2$.

$f\left(\bar{x}_{min}\right) = 0.000127276$, если $n = 10$.

То есть для каждого значения n надо пересчитывать значение глобального минимума.

График: Рисунок 2.10 нас 46 стр.

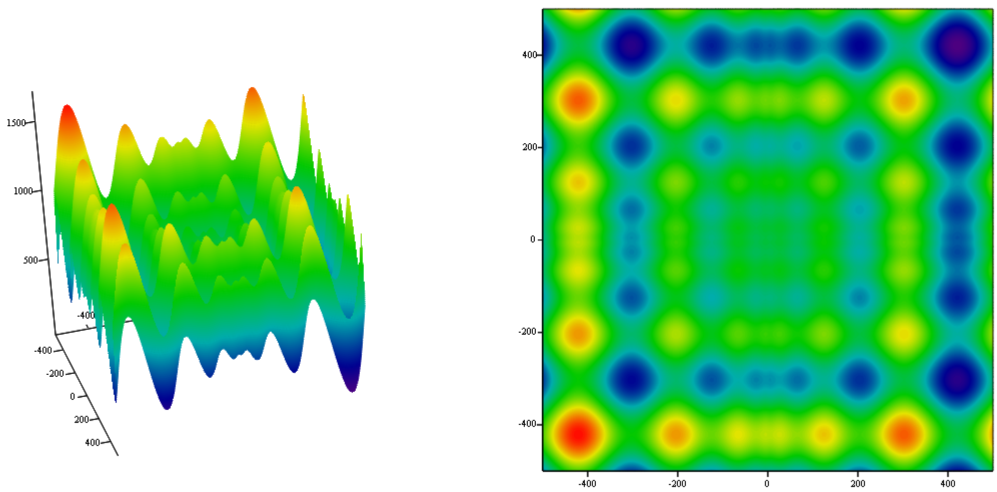


Рисунок 2.10. Функция Швефеля

2.10.2 Параметры для алгоритмов оптимизации

Точность вычислений:	$\varepsilon = 2.5.$
Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) :	$NumberOfParts_j = 4095$ $(j = \overline{1,n})$.
Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) :	$(k_2)_j = 12$ $(j = \overline{1,n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } \left(k_2\right)_j \in \mathbb{N}, \left(j = \overline{1,n}\right).$$

2.10.3 Основная задача и подзадачи

Изменяемый параметр:	n — размерность вещественного вектора.
Значение в основной задаче:	$n = 2.$
Подзадача №2:	$n = 3.$
Подзадача №3:	$n = 4.$
Подзадача №4:	$n = 5.$
Подзадача №5:	$n = 10.$
Подзадача №6:	$n = 20.$
Подзадача №7:	$n = 30.$

2.10.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно $(k = \overline{1,N})$. Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1,n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N}\left(\frac{\sqrt{\sum_{j=1}^{n}\left(\left(\bar{x}_{submin}^k\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n}\right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N}\left|f\left(\bar{x}_{submin}^k\right) - f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.10.5 Свойства задачи

Условной или безусловной оптимизации:	Задача безусловной оптимизации.
Одномерной или многомерной оптимизации:	Многомерной: n.
Функция унимодальная или многоэкстремальная:	Функция многоэкстремальная.
Функция стохастическая или нет:	Функция не стохастическая.
Особенности:	Для каждого значения n надо пересчитывать значение глобального минимума.

2.10.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.10. Код функции MHL_TestFunction_Schwefel

```
double double MHL_TestFunction_Schwefel(double *x, int VMHL_N)
{
/*
Функция многих переменных: функция Швефеля.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - указатель на исходный массив;
 VMHL_N - размер массива x.
Возвращаемое значение:
 Значение тестовой функции в точке x.
*/
double VMHL_Result=418.9829*VMHL_N;

for (int i=0;i<VMHL_N;i++)
    VMHL_Result -= x[i]*sin(sqrt(fabs(x[i])));

return VMHL_Result;
}
```

2.10.7 Ссылки

Данная функция приводится в следующих источниках:

1. [1, стр. 9] — Empirical review of standard benchmark functions using evolutionary global optimization.
2. [13] — Benchmark Problems.

Обратите внимание, что в англоязычном секторе часто данная функция дается с обозначением глобального минимума в точке $(\bar{x}_{min})_j = 1$ $(j = \overline{1,n})$, $f(\bar{x}_{min}) = 0$ (например, [14], [15]). Это не правильно: $f\begin{pmatrix}1\\1\end{pmatrix} = 836.28286$. Скорее всего в каком-то источнике вначале допустили ошибку, и она стала копироваться в остальные.

В некоторых источниках (например, [16], [5, стр. 7]) в формуле тестовой функции присутствует только второе слагаемое, а глобальный минимум определяется как $418.9829n$. Это неправильно, так как хоть два слагаемых похожи друг на друга (2.10), но они не дают строгий ноль в сумме, и при увеличении размерности различие между ними увеличивается.

2.11 Функция Step (модифицированная версия De Jong 3)

2.11.1 Описание функции

Идентификатор: MHL_TestFunction_StepFunction.

Наименование: Функция Step (модифицированная версия De Jong 3).

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = \begin{cases} \sum\limits_{i=1}^{n} \left(int\left(\bar{x}_i\right)\right)^2, \text{ если } \sum\limits_{i=1}^{n} \left|int\left(\bar{x}_i\right)\right| \neq 0; \\ \left(\sum\limits_{i=1}^{n} \left|\bar{x}_i\right|\right) - 1, \text{ иначе.} \end{cases}, \text{ где} \tag{2.11}$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = -5$, $Right_j = 5$, $j = \overline{1,n}$.

Обозначение: $\bar{x}$ — вещественный вектор;

n — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min} = \arg\min_{\bar{x} \in X} f(\bar{x})$.

Точка минимума: $\bar{x}_{min} = (0, 0, \ldots, 0)^{\mathrm{T}}$, то есть $(\bar{x}_{min})_j = 0$ $(j = \overline{1,n})$.

Минимум функции: $f(\bar{x}_{min}) = -1$.

График: Рисунок 2.11 нас 50 стр., 2.12 нас 50 стр.

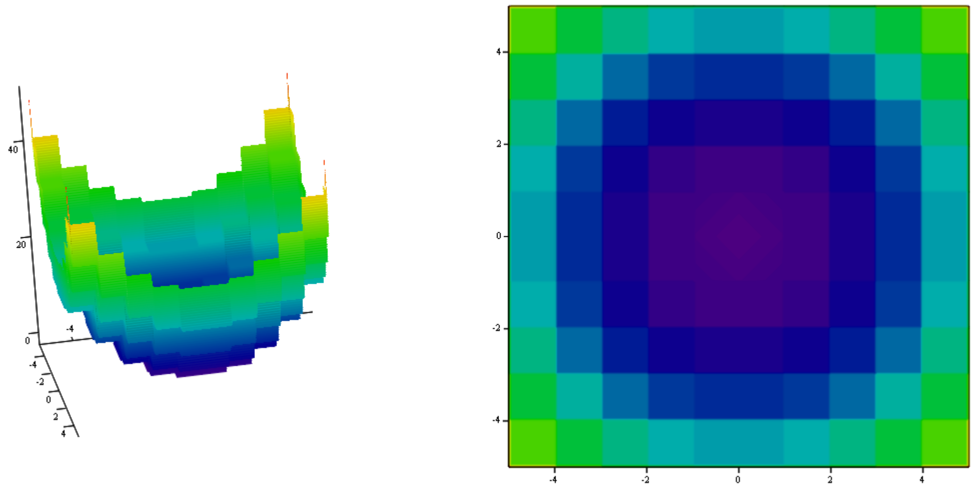


Рисунок 2.11. Функция Step (модифицированная версия De Jong 3)

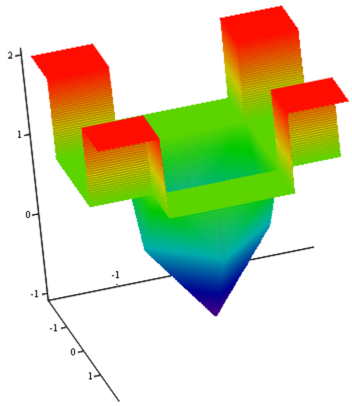


Рисунок 2.12. Функция Step (модифицированная версия De Jong 3) в области около точки минимума

2.11.2 Параметры для алгоритмов оптимизации

Точность вычислений: $\varepsilon = 0.025.$

Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) : $NumberOfParts_j = 4095 \ (j = \overline{1, n}).$

Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) : $(k_2)_j = 12 \ (j = \overline{1, n}).$

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } \ (k_2)_j \in \mathbb{N}, \left(j = \overline{1, n}\right).$$

2.11.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

Подзадача №2: $n = 3$.

Подзадача №3: $n = 4$.

Подзадача №4: $n = 5$.

Подзадача №5: $n = 10$.

Подзадача №6: $n = 20$.

Подзадача №7: $n = 30$.

2.11.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно ($k = \overline{1, N}$). Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1, n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N} \left(\frac{\sqrt{\sum_{j=1}^{n} \left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n} \right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N} \left|f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.11.5 Свойства задачи

Условной или безусловной оптимизации:	Задача безусловной оптимизации.
Одномерной или многомерной оптимизации:	Многомерной: n.
Функция унимодальная или многоэкстремальная:	Функция унимодальная.
Функция стохастическая или нет:	Функция не стохастическая.
Особенности:	На большей части множества допустимых решений производная функции равна нулю.

2.11.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.11. Код функции MHL_TestFunction_StepFunction

```
double MHL_TestFunction_StepFunction(double *x, int VMHL_N)
{
/*
Функция многих переменных: Функция Step (модифицированная версия De Jong 3).
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - указатель на исходный массив;
 VMHL_N - размер массива x.
Возвращаемое значение:
 Значение тестовой функции в точке x.
*/
    double VMHL_Result=0;

    double H=0;

    for (int i=0;i<VMHL_N;i++)
        H+=fabs(int(x[i]));

    if (H!=0)
    {
        for (int i=0;i<VMHL_N;i++)
            VMHL_Result+=(int(x[i]))*(int(x[i]));
    }
    else
    {
        for (int i=0;i<VMHL_N;i++)
            VMHL_Result+=fabs(x[i]);
    }

    return VMHL_Result;
}
```

2.11.7 Ссылки

Данная функция приводится в следующих источниках (без дополнительной добавки в области $\bar{x}_i \in (-1; 1), i = \overline{1, n}$):

1. [17, стр. 729] — International Conference on Intelligent Computing: Intelligent computing.

2.12 Аддитивная потенциальная функция

2.12.1 Описание функции

Идентификатор: MHL_TestFunction_AdditivePotential.

Наименование: Аддитивная потенциальная функция.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = z(\bar{x}_1) + z(\bar{x}_2), \text{ где} \tag{2.12}$$

$$z(v) = -\frac{1}{(v-1)^2 + 0.2} - \frac{1}{2(v-2)^2 + 0.15} - \frac{1}{3(v-3)^2 + 0.3},$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = 0$, $Right_j = 4$, $j = \overline{1, n}$, $n = 2$.

Обозначение: $\bar{x}$ — вещественный вектор;

$n = 2$ — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min} = \arg\min_{\bar{x} \in X} f(\bar{x})$.

Точка минимума: $\bar{x}_{min} = (2, 2)^{\mathrm{T}}$, то есть $(\bar{x}_{min})_j = 2$ $(j = \overline{1, n})$.

Минимум функции: $f(\bar{x}_{min}) = -15.606060606060606$.

График: Рисунок 2.13 нас 54 стр.

2.12.2 Параметры для алгоритмов оптимизации

Точность вычислений: $\varepsilon = 0.01$.

Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) : $NumberOfParts_j = 4095$ $(j = \overline{1, n})$.

Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) : $(k_2)_j = 12$ $(j = \overline{1, n})$.

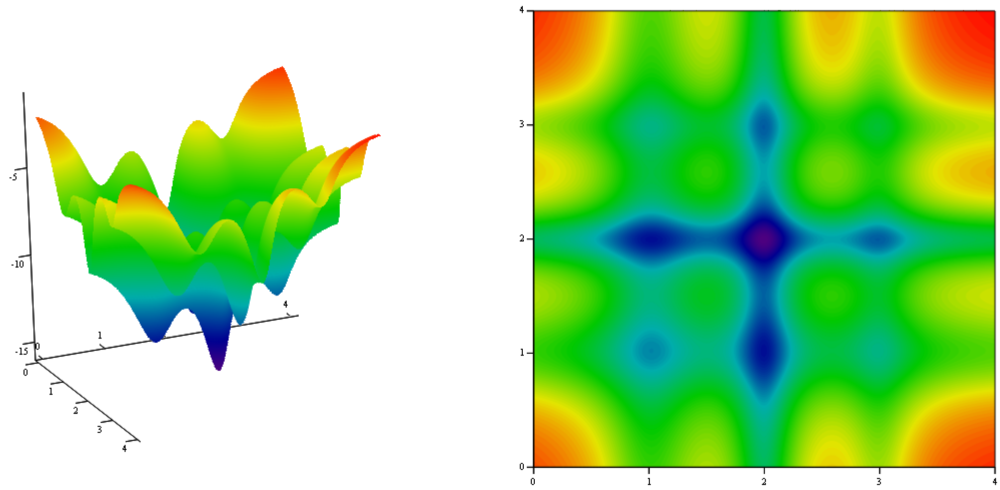


Рисунок 2.13. Аддитивная потенциальная функция

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } \left(k_2\right)_j \in \mathbb{N}, \left(j = \overline{1,n}\right).$$

2.12.3 Основная задача и подзадачи

Изменяемый параметр:	n — размерность вещественного вектора.
Значение в основной задаче:	$n = 2$.

2.12.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно ($k = \overline{1,N}$). Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1,n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N}\left(\frac{\sqrt{\sum_{j=1}^{n}\left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n}\right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N} \left| f\left(\bar{x}_{submin}^{k}\right) - f\left(\bar{x}_{min}\right) \right|}{N}.$$

2.12.5 Свойства задачи

Условной или безусловной оптимизации:	Задача безусловной оптимизации.
Одномерной или многомерной оптимизации:	Многомерной (двумерной).
Функция унимодальная или многоэкстремальная:	Функция многоэкстремальная.
Функция стохастическая или нет:	Функция не стохастическая.
Особенности:	Нет.

2.12.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.12. Код функции MHL_TestFunction_AdditivePotential

```
double MHL_TestFunction_AdditivePotential(double x, double y)
{
/*
Функция двух переменных: аддитивная потенциальная функция.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - первая вещественная переменная;
 y - вторая вещественная переменная.
Возвращаемое значение:
 Значение тестовой функции в точке (x,y).
*/
double VMHL_Result;
double z1=-(1./((x-1.)*(x-1.)+0.2))-(1./(2.*(x-2.)*(x-2.)+0.15))-(1./(3.*(x-3.)*(x
   -3.)+0.3));
double z2=-(1./((y-1.)*(y-1.)+0.2))-(1./(2.*(y-2.)*(y-2.)+0.15))-(1./(3.*(y-3.)*(y
   -3.)+0.3));
VMHL_Result=z1+z2;
return VMHL_Result;
}
```

2.12.7 Ссылки

Данная функция приводится в следующих источниках:

1. [18, стр. 33] — Эволюционные методы моделирования и оптимизации сложных систем.

2.13 Функция Egg Holder

2.13.1 Описание функции

Идентификатор: MHL_TestFunction_EggHolder.

Наименование: Функция Egg Holder.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right) = -\bar{x}_1 \sin\left(\sqrt{\left|\bar{x}_1 - 47 - \bar{x}_2\right|}\right) - \left(\bar{x}_2 + 47\right) \sin\left(\sqrt{\left|\frac{\bar{x}_1}{2} + 47 + \bar{x}_2\right|}\right), \text{ где} \quad (2.13)$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = -512$, $Right_j = 512$, $j = \overline{1, n}$, $n = 2$.

Обозначение: $\bar{x}$ — вещественный вектор;

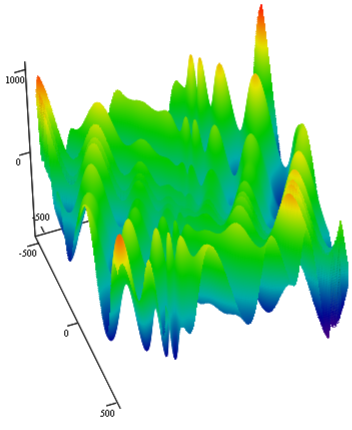
$n = 2$ — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min} = \arg\min_{\bar{x} \in X} f\left(\bar{x}\right)$.

Точка минимума: $\bar{x}_{min} = (512, 404.2319)^{\mathrm{T}}$.

Минимум функции: $f\left(\bar{x}_{min}\right) = -959.64067$.

График: Рисунок 2.14 нас 56 стр.



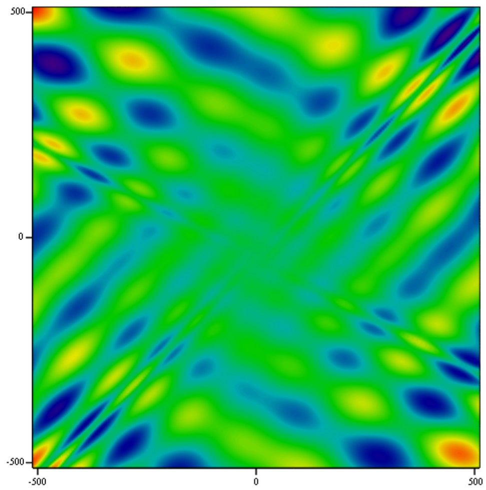


Рисунок 2.14. Функция Egg Holder

2.13.2 Параметры для алгоритмов оптимизации

Точность вычислений:	$\varepsilon = 2.5.$
Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) :	$NumberOfParts_j = 4095$ $(j = \overline{1,n})$.
Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) :	$(k_2)_j = 12$ $(j = \overline{1,n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } (k_2)_j \in \mathbb{N}, \left(j = \overline{1,n}\right).$$

2.13.3 Основная задача и подзадачи

Изменяемый параметр:	n — размерность вещественного вектора.
Значение в основной задаче:	$n = 2.$

2.13.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно $(k = \overline{1,N})$. Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1,n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N}\left(\frac{\sqrt{\sum_{j=1}^{n}\left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n}\right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N}\left|f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.13.5 Свойства задачи

Условной или безусловной оптимизации:	Задача безусловной оптимизации.
Одномерной или многомерной оптимизации:	Многомерной (двумерной).
Функция унимодальная или многоэкстремальная:	Функция многоэкстремальная.
Функция стохастическая или нет:	Функция не стохастическая.
Особенности:	Нет.

2.13.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.13. Код функции MHL_TestFunction_EggHolder

```
double MHL_TestFunction_EggHolder(double x, double y)
{
/*
Функция двух переменных: функция Egg Holder.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - первая вещественная переменная;
 y - вторая вещественная переменная.
Возвращаемое значение:
 Значение тестовой функции в точке (x,y).
*/
double VMHL_Result;

VMHL_Result=-x*sin(sqrt(fabs(x-(y+47.))))-(y+47)*sin(sqrt(fabs(x/2.+47+y)));

return VMHL_Result;
}
```

2.13.7 Ссылки

Данная функция приводится в следующих источниках:

1. [4, стр. 15] — A Literature Survey of Benchmark Functions For Global Optimization Problems.

2.14 Функция Химмельблау

2.14.1 Описание функции

Идентификатор: MHL_TestFunction_Himmelblau.

Наименование: Функция Химмельблау.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right)=\left(\bar{x}_1^2+\bar{x}_2-11\right)^2+\left(\bar{x}_1+\bar{x}^2-7\right)^2,\text{ где} \tag{2.14}$$

$\bar{x}\in X$, $\bar{x}_j\in[Left_j;Right_j]$, $Left_j=-5$, $Right_j=5$, $j=\overline{1,n}$, $n=2$.

Обозначение: $\bar{x}$ — вещественный вектор;

$n=2$ — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min}=\arg\min_{\bar{x}\in X} f\left(\bar{x}\right)$.

Точки минимума: $\bar{x}_{min}^1=\left(3,2\right)^{\mathrm{T}}$,

$\bar{x}_{min}^2\approx\left(-2.8051183,3.131312\right)^{\mathrm{T}}$

$\bar{x}_{min}^3\approx\left(-3.779310,-3.283186\right)^{\mathrm{T}}$

$\bar{x}_{min}^4\approx\left(3.584428,-1.848126\right)^{\mathrm{T}}$.

Минимум функции: $f\left(\bar{x}_{min}^i\right)=0$, $i=\overline{1,4}$.

График: Рисунок 2.15 нас 59 стр.

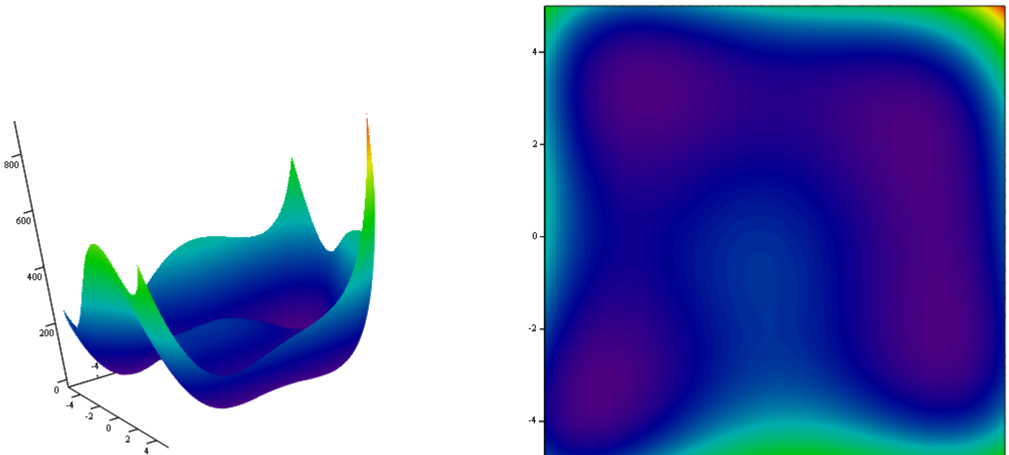


Рисунок 2.15. Функция Химмельблау

2.14.2 Параметры для алгоритмов оптимизации

Точность вычислений: $\varepsilon = 0.025$.

Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) : $NumberOfParts_j = 4095$ $(j = \overline{1, n})$.

Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) : $(k_2)_j = 12$ $(j = \overline{1, n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } (k_2)_j \in \mathbb{N}, \left(j = \overline{1, n}\right).$$

2.14.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

2.14.4 Нахождение ошибки оптимизации

Внимание! В отличии от других функций формулы нахождения ошибок другие, так как есть несколько идентичных по значению целевой функции глобальных минимумов.

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно $(k = \overline{1, N})$. Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}^1_{min}\right)_j\right| < \varepsilon, j = \overline{1, n}; \\ 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}^2_{min}\right)_j\right| < \varepsilon, j = \overline{1, n}; \\ 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}^3_{min}\right)_j\right| < \varepsilon, j = \overline{1, n}; \\ 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}^4_{min}\right)_j\right| < \varepsilon, j = \overline{1, n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \min_{i=\overline{1,4}} \left\{ \frac{\sum_{k=1}^{N} \left(\frac{\sqrt{\sum_{j=1}^{n} \left(\left(\bar{x}_{submin}^{k} \right)_j - \left(\bar{x}_{min}^{i} \right)_j \right)^2}}{n} \right)}{N} \right\}.$$

Ошибка по значениям целевой функции: (без изменений)

$$E_f = \frac{\sum_{k=1}^{N} \left| f\left(\bar{x}_{submin}^{k}\right) - f\left(\bar{x}_{min}\right) \right|}{N}.$$

2.14.5 Свойства задачи

Условной или безусловной оптимизации: Задача безусловной оптимизации.

Одномерной или многомерной оптимизации: Многомерной: (двумерной).

Функция унимодальная или многоэкстремальная: Функция многоэкстремальная.

Функция стохастическая или нет: Функция не стохастическая.

Особенности: Есть 4 глобальных минимума.

2.14.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.14. Код функции MHL_TestFunction_Himmelblau

```
double MHL_TestFunction_Himmelblau(double x, double y)
{
/*
Функция двух переменных: функция Химмельблау.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - первая вещественная переменная;
 y - вторая вещественная переменная.
Возвращаемое значение:
 Значение тестовой функции в точке (x,y).
*/
double VMHL_Result;
VMHL_Result=(x*x+y-11)*(x*x+y-11)+(x+y*y-7)*(x+y*y-7);
return VMHL_Result;
}
```

2.14.7 Ссылки

Данная функция приводится в следующих источниках:

1. [19] — Himmelblau's function.
2. [20] — Minimization of the Himmelblau Function.

2.15 Перевернутая функция Розенброка

2.15.1 Описание функции

Идентификатор:	MHL_TestFunction_InvertedRosenbrock.
Наименование:	Перевернутая функция Розенброка.
Тип:	Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = \frac{-100}{100\left(\bar{x}_1^2 - \bar{x}_2\right) + \left(1. - \bar{x}_1\right)^2 + 600}, \text{ где} \tag{2.15}$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = -5$, $Right_j = 5$, $j = \overline{1, n}$, $n = 2$.

Обозначение:	$\bar{x}$ — вещественный вектор;
	$n = 2$ — размерность вещественного вектора.
Решаемая задача оптимизации:	$\bar{x}_{min} = \arg\min_{\bar{x} \in X} f(\bar{x})$.
Точка минимума:	$\bar{x}_{min} = (0.00990099, 5)^{\mathrm{T}}$.
Минимум функции:	$f(\bar{x}_{min}) = -0.99019608$.
График:	Рисунок 2.16 нас 63 стр.

2.15.2 Параметры для алгоритмов оптимизации

Точность вычислений:	$\varepsilon = 0.025$.
Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) :	$NumberOfParts_j = 4095$ $(j = \overline{1, n})$.
Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) :	$(k_2)_j = 12$ $(j = \overline{1, n})$.

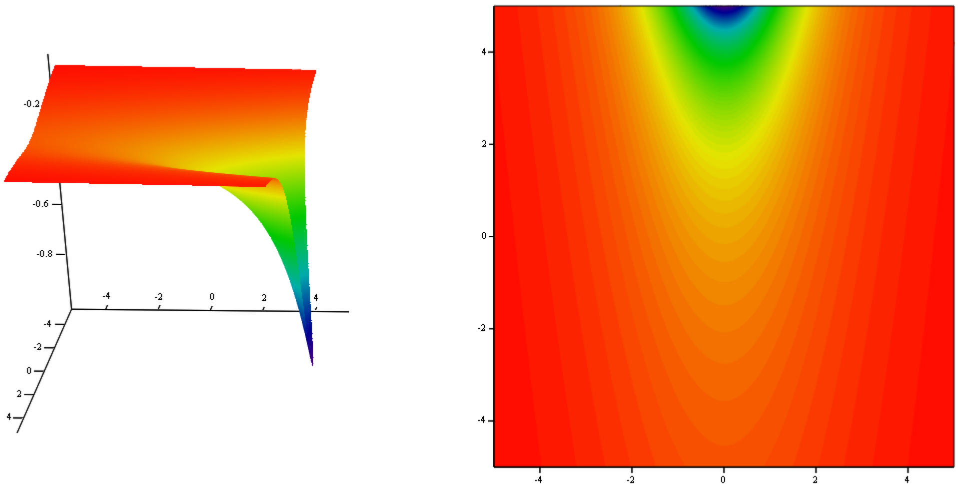


Рисунок 2.16. Перевернутая функция Розенброка

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } \ (k_2)_j \in \mathbb{N}, \left(j = \overline{1, n}\right).$$

2.15.3 Основная задача и подзадачи

Изменяемый параметр:	n — размерность вещественного вектора.
Значение в основной задаче:	$n = 2$.

2.15.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно ($k = \overline{1, N}$). Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1, n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N} \left(\frac{\sqrt{\sum_{j=1}^{n} \left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n} \right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N} \left| f\left(\bar{x}_{submin}^{k}\right) - f\left(\bar{x}_{min}\right) \right|}{N}.$$

2.15.5 Свойства задачи

Условной или безусловной оптимизации:	Задача безусловной оптимизации.
Одномерной или многомерной оптимизации:	Многомерной (двумерной).
Функция унимодальная или многоэкстремальная:	Функция унимодальная на рассматриваемой области.
Функция стохастическая или нет:	Функция не стохастическая.
Особенности:	Авторская модификация функции.

2.15.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.15. Код функции MHL_TestFunction_InvertedRosenbrock

```
double MHL_TestFunction_InvertedRosenbrock(double x, double y)
{
/*
Функция двух переменных: перевернутая функция Розенброка.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - первая вещественная переменная;
 y - вторая вещественная переменная.
Возвращаемое значение:
 Значение тестовой функции в точке (x,y).
*/
double VMHL_Result;

VMHL_Result=-100./(100.*(x*x-y)+(1.-x)*(1.-x)+600.);

return VMHL_Result;
}
```

2.15.7 Ссылки

Представленная здесь функция является модификацией неправильной записи функции из данного источника:

1. [18, стр. 29] — Эволюционные методы моделирования и оптимизации сложных систем.

2.16 Функция Катникова

2.16.1 Описание функции

Идентификатор: MHL_TestFunction_Katnikov.

Наименование: Функция Катникова.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = 0.5\left(\bar{x}_1^2 + \bar{x}_2^2\right)\left(2A + A\cos(1.5\bar{x}_1)\cos(3.14\bar{x}_2) + A\cos\left(\sqrt{5}\bar{x}_1\right)\cos(3.5\bar{x}_2)\right), \text{ где} \quad (2.16)$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = -5$, $Right_j = 5$, $j = \overline{1,n}$, $n = 2$, $A = 0.8$.

Обозначение: $\bar{x}$ — вещественный вектор;

$n = 2$ — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min} = \arg\min_{\bar{x} \in X} f(\bar{x})$.

Точка минимума: $\bar{x}_{min} = (0,0)^{\mathrm{T}}$, то есть $(\bar{x}_{min})_j = 0$ $(j = \overline{1,n})$.

Минимум функции: $f(\bar{x}_{min}) = 0$.

График: Рисунок 2.17 нас 65 стр.

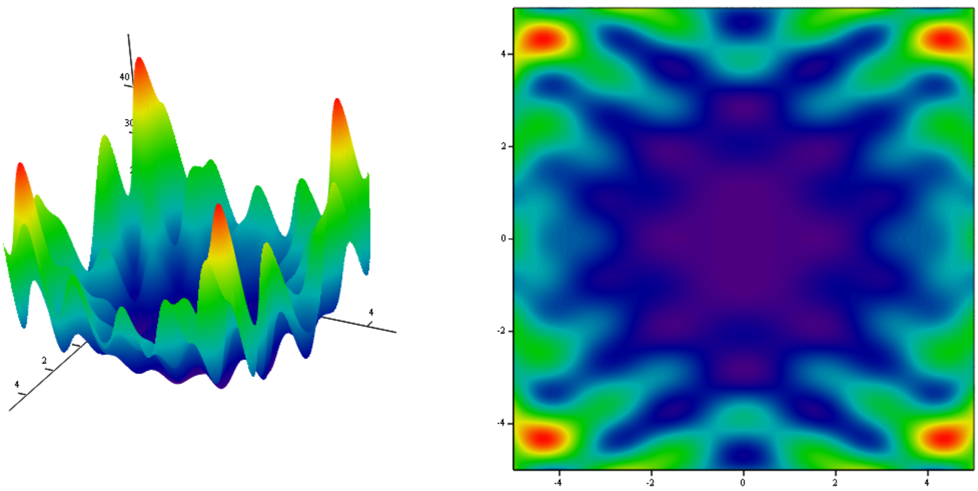


Рисунок 2.17. Функция Катникова

2.16.2 Параметры для алгоритмов оптимизации

Точность вычислений: $\varepsilon = 0.025$.

Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) : $NumberOfParts_j = 4095$ $(j = \overline{1, n})$.

Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) : $(k_2)_j = 12$ $(j = \overline{1, n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\,(Right_j - Left_j)}{\varepsilon}, \text{где } (k_2)_j \in \mathbb{N}, \left(j = \overline{1, n}\right).$$

2.16.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

2.16.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно $(k = \overline{1, N})$. Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1, n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N} \left(\frac{\sqrt{\sum_{j=1}^{n} \left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n} \right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N} \left| f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.16.5 Свойства задачи

Условной или безусловной оптимизации: Задача безусловной оптимизации.

Одномерной или многомерной оптимизации: Многомерной: (двумерной).

Функция унимодальная или многоэкстремальная: Функция многоэкстремальная.

Функция стохастическая или нет: Функция не стохастическая.

Особенности: Нет.

2.16.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.16. Код функции MHL_TestFunction_Katnikov

```
double MHL_TestFunction_Katnikov(double x, double y)
{
/*
Функция двух переменных: функция Катникова.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - первая вещественная переменная;
 y - вторая вещественная переменная.
Возвращаемое значение:
 Значение тестовой функции в точке (x,y).
*/
double VMHL_Result;
double A=0.8;
VMHL_Result=0.5*(x*x+y*y)*(2*A+A*cos(1.5*x)*cos(3.14*y)+A*cos(sqrt(5)*x)*cos(3.5*y));
return VMHL_Result;
}
```

2.16.7 Ссылки

Данная функция приводится в следующих источниках:

1. [18, стр. 31] — Эволюционные методы моделирования и оптимизации сложных систем.

2.17 Функция Multiextremal3

2.17.1 Описание функции

Идентификатор: MHL_TestFunction_Multiextremal3.

Наименование: Функция Multiextremal3.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = \bar{x}_1^2 \left|\sin(2\bar{x}_1)\right| + \bar{x}_2^2 \left|\sin(2\bar{x}_2)\right| - \frac{1}{5\bar{x}_1^2 + 5\bar{x}_2^2 + 0.2} + 5, \text{ где} \quad (2.17)$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = -5$, $Right_j = 5$, $j = \overline{1, n}$, $n = 2$.

Обозначение: $\bar{x}$ — вещественный вектор;

$n = 2$ — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min} = \arg\min_{\bar{x} \in X} f(\bar{x})$.

Точка минимума: $\bar{x}_{min} = (0, 0)^{\mathrm{T}}$, то есть $(\bar{x}_{min})_j = 0$ $(j = \overline{1, n})$.

Минимум функции: $f(\bar{x}_{min}) = 0$.

График: Рисунок 2.18 нас 68 стр.

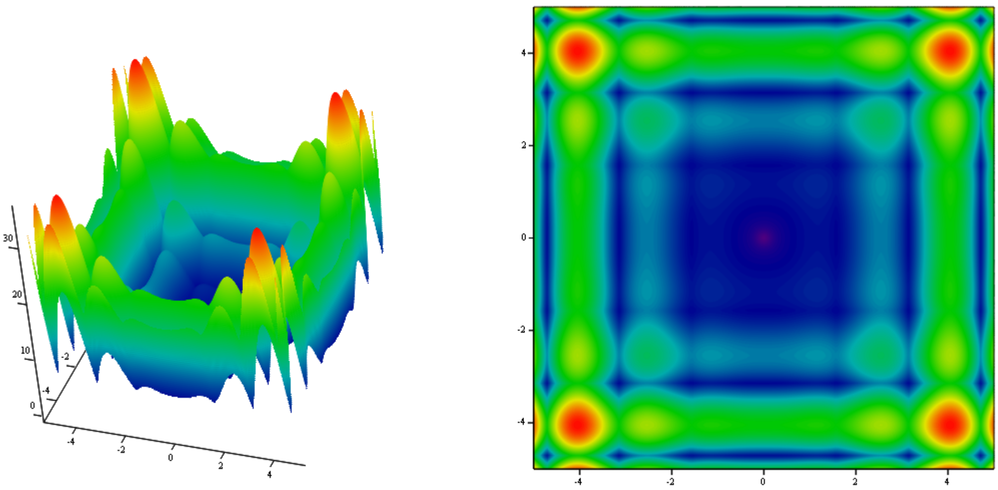


Рисунок 2.18. Функция Multiextremal3

2.17.2 Параметры для алгоритмов оптимизации

Точность вычислений:	$\varepsilon = 0.025$.
Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) :	$NumberOfParts_j = 4095$ $(j = \overline{1,n})$.
Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) :	$(k_2)_j = 12$ $(j = \overline{1,n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } (k_2)_j \in \mathbb{N}, \left(j = \overline{1,n}\right).$$

2.17.3 Основная задача и подзадачи

Изменяемый параметр:	n — размерность вещественного вектора.
Значение в основной задаче:	$n = 2$.

2.17.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно $(k = \overline{1,N})$. Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1,n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N}\left(\frac{\sqrt{\sum_{j=1}^{n}\left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n}\right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N}\left|f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.17.5 Свойства задачи

Условной или безусловной оптимизации: Задача безусловной оптимизации.

Одномерной или многомерной оптимизации: Многомерной: (двумерной).

Функция унимодальная или многоэкстремальная: Функция многоэкстремальная.

Функция стохастическая или нет: Функция не стохастическая.

Особенности: Нет.

2.17.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.17. Код функции MHL_TestFunction_Multiextremal3

```
double MHL_TestFunction_Multiextremal3(double x, double y)
{
/*
Функция двух переменных: функция Multiextremal3.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - первая вещественная переменная;
 y - вторая вещественная переменная.
Возвращаемое значение:
 Значение тестовой функции в точке (x,y).
*/
double VMHL_Result;
VMHL_Result=x*x*fabs(sin(2.*x))+y*y*fabs(sin(2.*y))-1./(5.*x*x+5.*y*y+0.2)+5.;
return VMHL_Result;
}
```

2.17.7 Ссылки

Данная функция приводится в следующих источниках:

1. [18, стр. 31] — Эволюционные методы моделирования и оптимизации сложных систем.

2.18 Функция Multiextremal4

2.18.1 Описание функции

Идентификатор: MHL_TestFunction_Multiextremal4.

Наименование: Функция Multiextremal4.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right) = 0.5\left(\bar{x}_1^2 + \bar{x}_1\bar{x}_2 + \bar{x}_2^2\right)\left(1 + 0.5\cos\left(1.5\bar{x}_1\right)\cos\left(3.2\bar{x}_1\bar{x}_2\right)\cos\left(3.14\bar{x}_2\right) + \right.$$
$$\left. + 0.5\cos\left(2.2\bar{x}_1\right)\cos\left(4.8\bar{x}_1\bar{x}_2\right)\cos\left(3.5\bar{x}_2\right)\right),\ \text{где} \qquad (2.18)$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = 0$, $Right_j = 4$, $j = \overline{1,n}$, $n = 2$.

Обозначение: $\bar{x}$ — вещественный вектор;

$n = 2$ — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min} = \arg\min_{\bar{x}\in X} f\left(\bar{x}\right)$.

Точка минимума: $\bar{x}_{min} = (0,0)^{\mathrm{T}}$, то есть $\left(\bar{x}_{min}\right)_j = 0$ $(j = \overline{1,n})$.

Минимум функции: $f\left(\bar{x}_{min}\right) = 0$.

График: Рисунок 2.19 на с 71 стр.

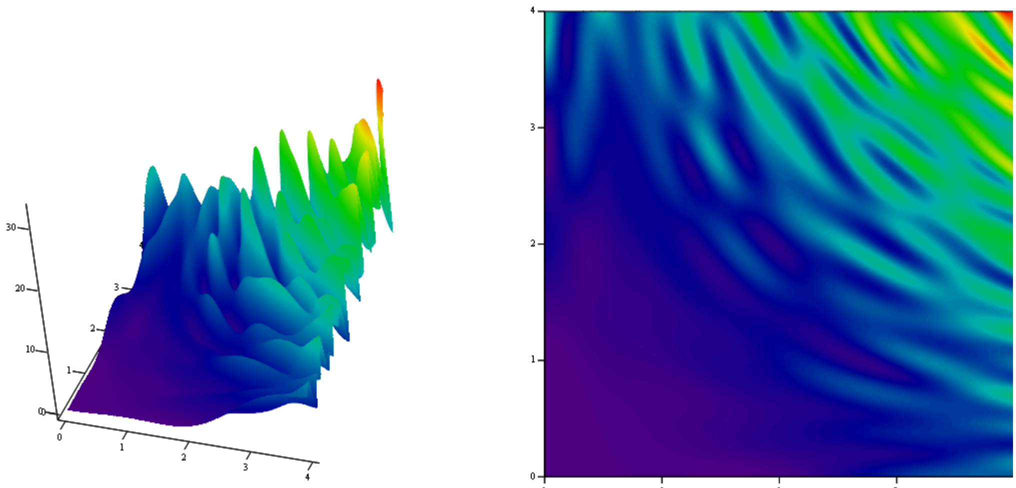


Рисунок 2.19. Функция Multiextremal4

2.18.2 Параметры для алгоритмов оптимизации

Точность вычислений:	$\varepsilon = 0.01$.
Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) :	$NumberOfParts_j = 4095$ $(j = \overline{1,n})$.
Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) :	$(k_2)_j = 12$ $(j = \overline{1,n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } (k_2)_j \in \mathbb{N}, \left(j = \overline{1,n}\right).$$

2.18.3 Основная задача и подзадачи

Изменяемый параметр:	n — размерность вещественного вектора.
Значение в основной задаче:	$n = 2$.

2.18.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно $(k = \overline{1,N})$. Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1,n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N} \left(\frac{\sqrt{\sum_{j=1}^{n} \left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n} \right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N} \left| f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right) \right|}{N}.$$

2.18.5 Свойства задачи

Условной или безусловной оптимизации: Задача безусловной оптимизации.

Одномерной или многомерной оптимизации: Многомерной: (двумерной).

Функция унимодальная или многоэкстремальная: Функция многоэкстремальная.

Функция стохастическая или нет: Функция не стохастическая.

Особенности: Нет.

2.18.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.18. Код функции MHL_TestFunction_Multiextremal4

```
double MHL_TestFunction_Multiextremal4(double x, double y)
{
/*
Функция двух переменных: функция Multiextremal4.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - первая вещественная переменная;
 y - вторая вещественная переменная.
Возвращаемое значение:
 Значение тестовой функции в точке (x,y).
*/
double VMHL_Result;
VMHL_Result=0.5*(x*x+x*y+y*y)*(1.+0.5*cos(1.5*x)*cos(3.2*x*y)*cos(3.14*y)+0.5*cos
    (2.2*x)*cos(4.8*x*y)*cos(3.5*y));
return VMHL_Result;
}
```

2.18.7 Ссылки

Данная функция приводится в следующих источниках:

1. [18, стр. 31] — Эволюционные методы моделирования и оптимизации сложных систем.

2.19 Мультипликативная потенциальная функция

2.19.1 Описание функции

Идентификатор: MHL_TestFunction_MultiplicativePotential.

Наименование: Мультипликативная потенциальная функция.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = -z(\bar{x}_1) \cdot z(\bar{x}_2), \text{ где} \tag{2.19}$$

$$z(v) = -\frac{1}{(v-1)^2 + 0.2} - \frac{1}{2(v-2)^2 + 0.15} - \frac{1}{3(v-3)^2 + 0.3},$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = 0$, $Right_j = 4$, $j = \overline{1, n}$, $n = 2$.

Обозначение: $\bar{x}$ — вещественный вектор;

$n = 2$ — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min} = \arg\min_{\bar{x} \in X} f(\bar{x})$.

Точка минимума: $\bar{x}_{min} = (2, 2)^{\mathrm{T}}$, то есть $(\bar{x}_{min})_j = 2$ $(j = \overline{1, n})$.

Минимум функции: $f(\bar{x}_{min}) = -60.8872819100091$.

График: Рисунок 2.20 нас 74 стр.

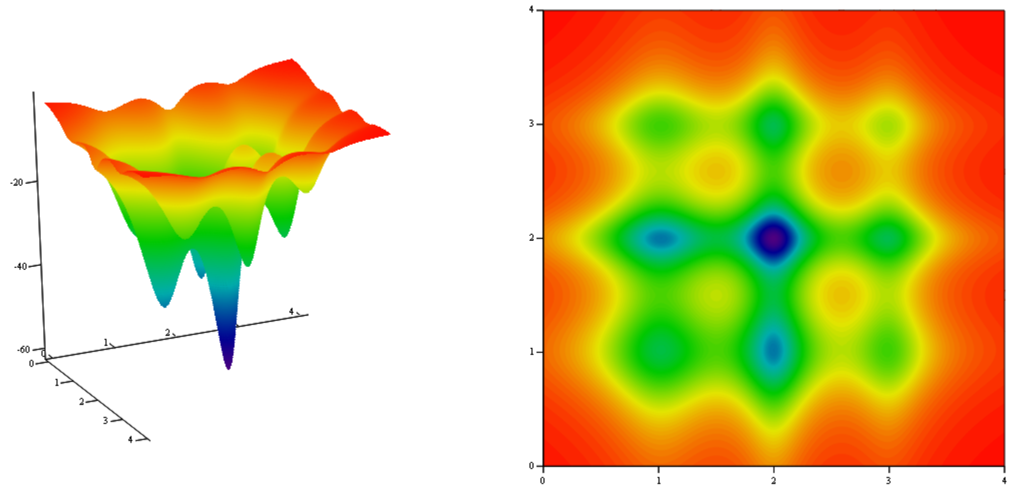


Рисунок 2.20. Мультипликативная потенциальная функция

2.19.2 Параметры для алгоритмов оптимизации

Точность вычислений: $\varepsilon = 0.01$.

Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) : $NumberOfParts_j = 4095$ $(j = \overline{1,n})$.

Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) : $(k_2)_j = 12$ $(j = \overline{1,n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } \left(k_2\right)_j \in \mathbb{N}, \left(j = \overline{1,n}\right).$$

2.19.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

2.19.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно $(k = \overline{1,N})$. Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1,n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N}\left(\frac{\sqrt{\sum_{j=1}^{n}\left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n}\right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N}\left|f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.19.5 Свойства задачи

Условной или безусловной оптимизации: Задача безусловной оптимизации.

Одномерной или многомерной оптимизации: Многомерной: (двумерной).

Функция унимодальная или многоэкстремальная: Функция многоэкстремальная.

Функция стохастическая или нет: Функция не стохастическая.

Особенности: Нет.

2.19.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.19. Код функции MHL_TestFunction_MultiplicativePotential

```
double MHL_TestFunction_MultiplicativePotential(double x, double y)
{
/*
Функция двух переменных: мультипликативная потенциальная функция.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - первая вещественная переменная;
 y - вторая вещественная переменная.
Возвращаемое значение:
 Значение тестовой функции в точке (x,y).
*/
double VMHL_Result;
double z1=-(1./((x-1.)*(x-1.)+0.2))-(1./(2.*(x-2.)*(x-2.)+0.15))-(1./(3.*(x-3.)*(x
    -3.)+0.3));
double z2=-(1./((y-1.)*(y-1.)+0.2))-(1./(2.*(y-2.)*(y-2.)+0.15))-(1./(3.*(y-3.)*(y
    -3.)+0.3));
VMHL_Result=-z1*z2;
return VMHL_Result;
}
```

2.19.7 Ссылки

Данная функция приводится в следующих источниках:

1. [18, стр. 32] — Эволюционные методы моделирования и оптимизации сложных систем.

2.20 Функция Rana

2.20.1 Описание функции

Идентификатор: MHL_TestFunction_Rana.

Наименование: Функция Rana.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right) = \bar{x}_1 \sin\left(\sqrt{\left|\bar{x}_2 + 1 - \bar{x}_1\right|}\right) \cos\left(\sqrt{\left|\bar{x}_2 + 1 + \bar{x}_1\right|}\right) + \tag{2.20}$$

$$+ \left(\bar{x}_2 + 1\right) \cos\left(\sqrt{\left|\bar{x}_2 + 1 - \bar{x}_1\right|}\right) \sin\left(\sqrt{\left|\bar{x}_2 + 1 + x\right|}\right), \text{ где} \tag{2.21}$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = -512$, $Right_j = 512$, $j = \overline{1,n}$, $n = 2$.

Обозначение: $\bar{x}$ — вещественный вектор;

$n = 2$ — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min} = \arg\min_{\bar{x} \in X} f\left(\bar{x}\right)$.

Точка минимума: $\bar{x}_{min} = \left(-488.6326, 512\right)^{\mathrm{T}}$.

Минимум функции: $f\left(\bar{x}_{min}\right) = -511.7328819$.

График: Рисунок 2.21 нас 77 стр.

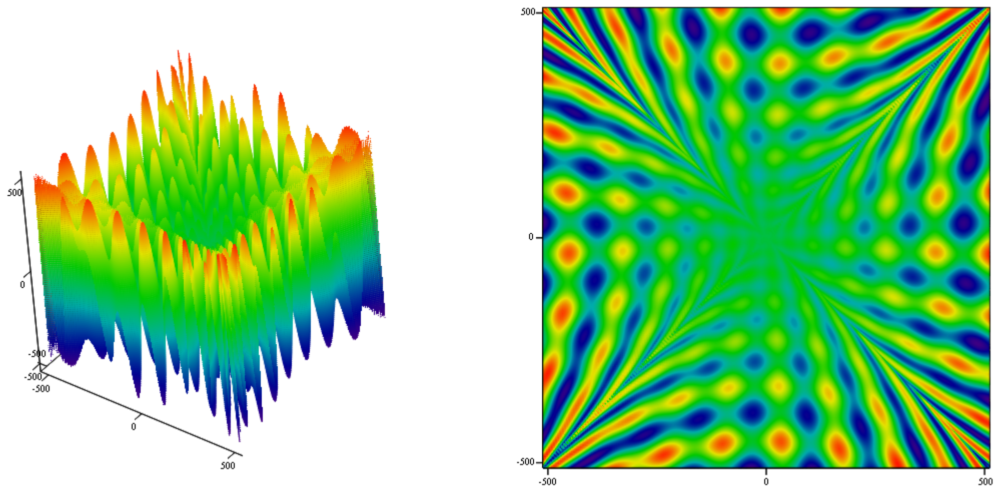


Рисунок 2.21. Функция Rana

2.20.2 Параметры для алгоритмов оптимизации

Точность вычислений: $\varepsilon = 2.5$.

Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) : $NumberOfParts_j = 4095$ $(j = \overline{1,n})$.

Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) : $(k_2)_j = 12$ $(j = \overline{1,n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } (k_2)_j \in \mathbb{N}, \left(j = \overline{1,n}\right).$$

2.20.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

2.20.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно $(k = \overline{1,N})$. Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1,n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N}\left(\frac{\sqrt{\sum_{j=1}^{n}\left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n}\right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N}\left|f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.20.5 Свойства задачи

Условной или безусловной оптимизации: Задача безусловной оптимизации.

Одномерной или многомерной оптимизации: Многомерной: (двумерной).

Функция унимодальная или многоэкстремальная: Функция многоэкстремальная.

Функция стохастическая или нет: Функция не стохастическая.

Особенности: Нет.

2.20.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.20. Код функции MHL_TestFunction_Rana

```
double MHL_TestFunction_Rana(double x, double y)
{
/*
Функция двух переменных: функция Rana.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - первая вещественная переменная;
 y - вторая вещественная переменная.
Возвращаемое значение:
 Значение тестовой функции в точке (x,y).
*/
double VMHL_Result;
VMHL_Result=x*sin(sqrt(fabs(y+1.-x)))*cos(sqrt(fabs(y+1.+x))) + (y+1.)*cos(sqrt(fabs(
    y+1.-x)))*sin(sqrt(fabs(y+1.+x)));
return VMHL_Result;
}
```

2.20.7 Ссылки

Данная функция приводится в следующих источниках:

1. [21] — Rana.
2. [8] — Non-linear Continuous Multi-Extremal Optimization.

2.21 Функция Растригина с изменением коэффициентов

2.21.1 Описание функции

Идентификатор: MHL_TestFunction_RastriginWithChange.

Наименование: Функция Растригина с изменением коэффициентов.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right)=0.1\bar{x}_1^2+0.1\bar{x}_2^2-4\cos\left(0.8\bar{x}_1\right)-4\cos\left(0.8\bar{x}_2\right)+8,\text{ где} \tag{2.22}$$

$\bar{x}\in X$, $\bar{x}_j\in\left[Left_j;Right_j\right]$, $Left_j=-16$, $Right_j=16$, $j=\overline{1,n}$, $n=2$.

Обозначение: $\bar{x}$ — вещественный вектор;

$n=2$ — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{max}=\arg\max_{\bar{x}\in X}f\left(\bar{x}\right)$.

Точка максимума: $\bar{x}_{max}=\left(0,0\right)^{\mathrm{T}}$, то есть $\left(\bar{x}_{max}\right)_j=0$ $(j=\overline{1,n})$.

Максимум функции: $f\left(\bar{x}_{max}\right)=0$.

График: Рисунок 2.22 нас 80 стр.

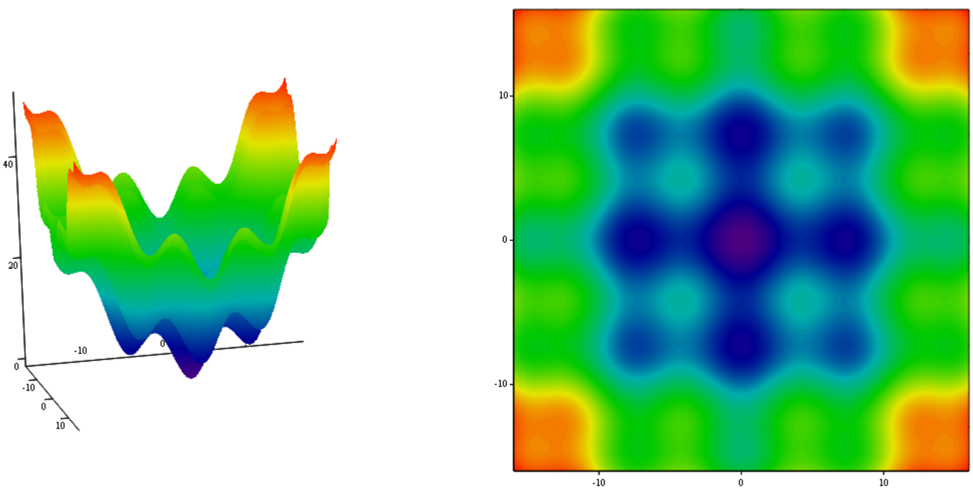


Рисунок 2.22. Функция Растригина с изменением коэффициентов

2.21.2 Параметры для алгоритмов оптимизации

Точность вычислений: $\varepsilon = 0.08$.

Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) : $NumberOfParts_j = 4095$ $(j = \overline{1,n})$.

Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) : $(k_2)_j = 12$ $(j = \overline{1,n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } (k_2)_j \in \mathbb{N}, \left(j = \overline{1,n}\right).$$

2.21.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

2.21.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно $(k = \overline{1,N})$. Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1,n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N}\left(\frac{\sqrt{\sum_{j=1}^{n}\left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n}\right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N}\left|f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.21.5 Свойства задачи

Условной или безусловной оптимизации:	Задача безусловной оптимизации.
Одномерной или многомерной оптимизации:	Многомерной: (двумерной).
Функция унимодальная или многоэкстремальная:	Функция многоэкстремальная.
Функция стохастическая или нет:	Функция не стохастическая.
Особенности:	Нет.

2.21.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.21. Код функции MHL_TestFunction_RastriginWithChange

```
double MHL_TestFunction_RastriginWithChange(double x, double y)
{
/*
Функция двух переменных: функция Растригина с изменением коэффициентов.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - первая вещественная переменная;
 y - вторая вещественная переменная.
Возвращаемое значение:
 Значение тестовой функции в точке (x,y).
*/
double VMHL_Result;

VMHL_Result=0.1*x*x+0.1*y*y-4.*cos(0.8*x)-4.*cos(0.8*y)+8.;

return VMHL_Result;
}
```

2.21.7 Ссылки

Данная функция приводится в следующих источниках:

1. [18, стр. 27] — Эволюционные методы моделирования и оптимизации сложных систем.

2.22 Функция Растригина овражная с поворотом осей

2.22.1 Описание функции

Идентификатор: MHL_TestFunction_RastriginWithTurning.

Наименование: Функция Растригина овражная с поворотом осей.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$\begin{aligned} f(\bar{x}) = &(0.1K_{\bar{x}_1}A(\bar{x}_1,\bar{x}_2))^2 + (0.1K_{\bar{x}_2}B(\bar{x}_1,\bar{x}_2))^2 - \\ &- 4\cos(0.8K_{\bar{x}_1}A(\bar{x}_1,\bar{x}_2)) - 4\cos(0.8K_{\bar{x}_2}B(\bar{x}_1,\bar{x}_2)) + 8, \text{ где} \\ &A(\bar{x}_1,\bar{x}_2) = \bar{x}_1\cos(\alpha) - \bar{x}_2\sin(\alpha), \\ &B(\bar{x}_1,\bar{x}_2) = \bar{x}_1\sin(\alpha) + \bar{x}_2\cos(\alpha), \end{aligned} \tag{2.23}$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = -16$, $Right_j = 16$, $j = \overline{1,n}$, $n = 2$, $K_{\bar{x}_1} = 1.5$, $K_{\bar{x}_2} = 0.8$, $\alpha = \frac{\pi}{2}$.

Обозначение: $\bar{x}$ — вещественный вектор;

$n = 2$ — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{max} = \arg\max_{\bar{x}\in X} f(\bar{x})$.

Точка максимума: $\bar{x}_{max} = (0,0)^{\mathrm{T}}$, то есть $(\bar{x}_{max})_j = 0$ ($j = \overline{1,n}$).

Максимум функции: $f(\bar{x}_{max}) = 0$.

График: Рисунок 2.23 нас 83 стр.

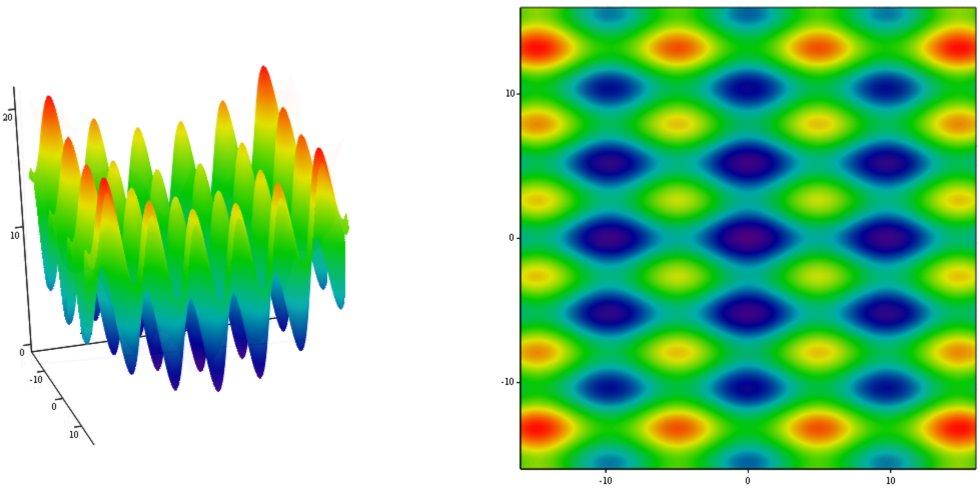


Рисунок 2.23. Функция Растригина овражная с поворотом осей

2.22.2 Параметры для алгоритмов оптимизации

Точность вычислений: $\varepsilon = 0.08$.

Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) : $NumberOfParts_j = 4095$ $(j = \overline{1,n})$.

Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) : $(k_2)_j = 12$ $(j = \overline{1,n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } (k_2)_j \in \mathbb{N}, \left(j = \overline{1,n}\right).$$

2.22.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

2.22.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно ($k = \overline{1,N}$). Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^N S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1,n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^N \left(\frac{\sqrt{\sum_{j=1}^n \left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n}\right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^N \left|f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.22.5 Свойства задачи

Условной или безусловной оптимизации: Задача безусловной оптимизации.

Одномерной или многомерной оптимизации: Многомерной: (двумерной).

Функция унимодальная или многоэкстремальная: Функция многоэкстремальная.

Функция стохастическая или нет: Функция не стохастическая.

Особенности: Нет.

2.22.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.22. Код функции MHL_TestFunction_RastriginWithTurning

```
double MHL_TestFunction_RastriginWithTurning(double x, double y)
{
/*
Функция двух переменных: функция Растригина овражная с поворотом осей.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - первая вещественная переменная;
 y - вторая вещественная переменная.
Возвращаемое значение:
 Значение тестовой функции в точке (x,y).
*/
double VMHL_Result;

double alpha=MHL_PI_2;
double kx=1.5;
double ky=0.8;

double A,B;
A=x*cos(alpha)-y*sin(alpha);
B=x*sin(alpha)+y*cos(alpha);

VMHL_Result=(0.1*kx*A)*(0.1*kx*A)+(0.1*ky*B)*(0.1*ky*B)-4.*cos(0.8*kx*A)-4.*cos(0.8*
    ky*B)+8.;

return VMHL_Result;
}
```

2.22.7 Ссылки

Данная функция приводится в следующих источниках:

1. [18, стр. 28] — Эволюционные методы моделирования и оптимизации сложных систем.

2.23 Функция ReverseGriewank

2.23.1 Описание функции

Идентификатор: MHL_TestFunction_ReverseGriewank.

Наименование: Функция ReverseGriewank.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right)=\frac{1}{\dfrac{\bar{x}_1^2+\bar{x}_2^2}{200}-cos\left(\bar{x}_0\right)cos\left(\dfrac{\bar{x}_2}{\sqrt{2}}\right)+2},\text{ где} \tag{2.24}$$

$\bar{x}\in X$, $\bar{x}_j\in[Left_j;Right_j]$, $Left_j=-10$, $Right_j=10$, $j=\overline{1,n}$, $n=2$.

Обозначение: $\bar{x}$ — вещественный вектор;

$n=2$ — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{max}=\arg\max_{\bar{x}\in X} f\left(\bar{x}\right)$.

Точка максимума: $\bar{x}_{max}=\left(0,0\right)^{\mathrm{T}}$, то есть $\left(\bar{x}_{max}\right)_j=0$ ($j=\overline{1,n}$).

Максимум функции: $f\left(\bar{x}_{max}\right)=1$.

График: Рисунок 2.24 на 86 стр.

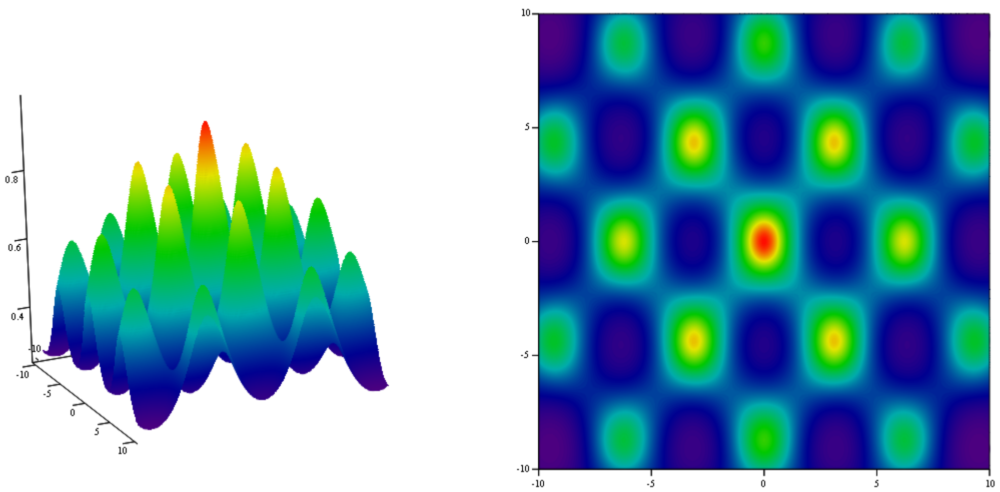


Рисунок 2.24. Функция ReverseGriewank

2.23.2 Параметры для алгоритмов оптимизации

Точность вычислений:	$\varepsilon = 0.05$.
Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) :	$NumberOfParts_j = 4095$ $(j = \overline{1,n})$.
Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) :	$(k_2)_j = 12$ $(j = \overline{1,n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } (k_2)_j \in \mathbb{N}, \left(j = \overline{1,n}\right).$$

2.23.3 Основная задача и подзадачи

Изменяемый параметр:	n — размерность вещественного вектора.
Значение в основной задаче:	$n = 2$.

2.23.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно ($k = \overline{1,N}$). Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1,n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N}\left(\frac{\sqrt{\sum_{j=1}^{n}\left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n}\right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N}\left|f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.23.5 Свойства задачи

Условной или безусловной оптимизации: Задача безусловной оптимизации.

Одномерной или многомерной оптимизации: Многомерной: (двумерной).

Функция унимодальная или многоэкстремальная: Функция многоэкстремальная.

Функция стохастическая или нет: Функция не стохастическая.

Особенности: Нет.

2.23.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.23. Код функции MHL_TestFunction_ReverseGriewank

```
double MHL_TestFunction_ReverseGriewank(double x, double y)
{
/*
Функция двух переменных: функция ReverseGriewank.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - первая вещественная переменная;
 y - вторая вещественная переменная.
Возвращаемое значение:
 Значение тестовой функции в точке (x,y).
*/
double VMHL_Result;

VMHL_Result = 1./((x*x+y*y)/200.-cos(x)*cos(y/sqrt(2.))+2.);

return VMHL_Result;
}
```

2.23.7 Ссылки

Так и не смог найти нормальный источник для этой функции. По внешнему виду похожа на функцию Гриванка, которую возвели в -1 степень. Откуда-то у меня находится со студенческих времен.

2.24 Функция «Лисьи норы» Шекеля

2.24.1 Описание функции

Идентификатор: MHL_TestFunction_ShekelsFoxholes.

Наименование: Функция «Лисьи норы» Шекеля.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right) = \frac{1}{\frac{1}{K} + \sum_{j=1}^{25} \frac{1}{j + \left(\bar{x}_1 - A_{1,j}\right)^6 + \left(\bar{x}_2 - A_{2,j}\right)^6}}, \text{ где} \tag{2.25}$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = -50$, $Right_j = 50$, $j = \overline{1,n}$, $n = 2$, $K = 500$,

$$A = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & -16 & 0 & 16 & 32 & -32 & -16 & 0 & 16 & 32 & -32 & -16 & 0 & 16 & 32 & -32 & -16 & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & -16 & -16 & -16 & -16 & 0 & 0 & 0 & 0 & 0 & 16 & 16 & 16 & 16 & 16 & 32 & 32 & 32 & 32 & 32 \end{pmatrix}.$$

Обозначение: $\bar{x}$ — вещественный вектор;

$n = 2$ — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min} = \arg\min_{\bar{x} \in X} f\left(\bar{x}\right)$.

Точка минимума: $\bar{x}_{min} = (-32, -32)^{\mathrm{T}}$, то есть $\left(\bar{x}_{min}\right)_j = -32$ $(j = \overline{1,n})$.

Минимум функции: $f\left(\bar{x}_{min}\right) = 0.99800384$.

График: Рисунок 2.25 нас 89 стр.

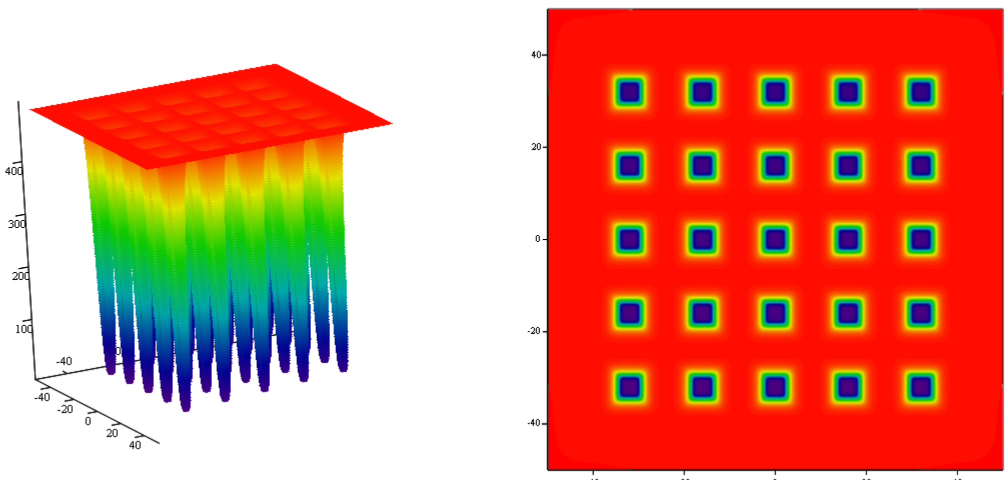


Рисунок 2.25. Функция «Лисьи норы» Шекеля

2.24.2 Параметры для алгоритмов оптимизации

Точность вычислений:	$\varepsilon = 0.25$.
Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) :	$NumberOfParts_j = 4095\ (j = \overline{1,n})$.
Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) :	$(k_2)_j = 12\ (j = \overline{1,n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\,(Right_j - Left_j)}{\varepsilon}, \text{где } (k_2)_j \in \mathbb{N}, \left(j = \overline{1,n}\right).$$

2.24.3 Основная задача и подзадачи

Изменяемый параметр:	n — размерность вещественного вектора.
Значение в основной задаче:	$n = 2$.

2.24.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно ($k = \overline{1,N}$). Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1,n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N} \left(\frac{\sqrt{\sum_{j=1}^{n} \left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n} \right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N} \left| f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right) \right|}{N}.$$

2.24.5 Свойства задачи

Условной или безусловной оптимизации: Задача безусловной оптимизации.

Одномерной или многомерной оптимизации: Многомерной: (двумерной).

Функция унимодальная или многоэкстремальная: Функция многоэкстремальная.

Функция стохастическая или нет: Функция не стохастическая.

Особенности: Глобальный минимум слабо отличается от локальных. Из локальных минимумов алгоритмам обычно сложно выбраться.

2.24.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.24. Код функции MHL_TestFunction_ShekelsFoxholes

```
double MHL_TestFunction_ShekelsFoxholes(double x, double y)
{
/*
Функция двух переменных: функция "Лисьи норы" Шекеля.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - первая вещественная переменная;
 y - вторая вещественная переменная.
Возвращаемое значение:
 Значение тестовой функции в точке (x,y).
*/
double VMHL_Result;
double K=500.;
double f1,f2;
int j,k;
int a[2][25];

a[0][0]=-32;
a[0][1]=-16;
a[0][2]=0;
a[0][3]=16;
a[0][4]=32;
a[0][5]=-32;
a[0][6]=-16;
a[0][7]=0;
a[0][8]=16;
a[0][9]=32;
a[0][10]=-32;
a[0][11]=-16;
a[0][12]=0;
a[0][13]=16;
a[0][14]=32;
```

```
a[0][15]=-32;
a[0][16]=-16;
a[0][17]=0;
a[0][18]=16;
a[0][19]=32;
a[0][20]=-32;
a[0][21]=-16;
a[0][22]=0;
a[0][23]=16;
a[0][24]=32;

a[1][0]=-32;
a[1][1]=-32;
a[1][2]=-32;
a[1][3]=-32;
a[1][4]=-32;
a[1][5]=-16;
a[1][6]=-16;
a[1][7]=-16;
a[1][8]=-16;
a[1][9]=-16;
a[1][10]=0;
a[1][11]=0;
a[1][12]=0;
a[1][13]=0;
a[1][14]=0;
a[1][15]=16;
a[1][16]=16;
a[1][17]=16;
a[1][18]=16;
a[1][19]=16;
a[1][20]=32;
a[1][21]=32;
a[1][22]=32;
a[1][23]=32;
a[1][24]=32;

VMHL_Result=1./K;
for (j=0;j<25;j++)
 {
 f1=1;
 for (k=0;k<6;k++) f1=f1*(x-a[0][j]);
 f2=1;
 for (k=0;k<6;k++) f2=f2*(y-a[1][j]);
 VMHL_Result+=1./(j+1.+f1+f2);
 }

VMHL_Result=1./VMHL_Result;

return VMHL_Result;
}
```

2.24.7 Ссылки

Данная функция приводится в следующих источниках:

1. [18, стр. 34] — Эволюционные методы моделирования и оптимизации сложных систем.

2. [17, стр. 729] — International Conference on Intelligent Computing: Intelligent computing.

2.25 Функция Сомбреро

2.25.1 Описание функции

Идентификатор: MHL_TestFunction_Sombrero.

Наименование: Функция Сомбреро.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right) = \frac{1 - sin\left(\sqrt{\bar{x}_1^2 + \bar{x}_2^2}\right)^2}{1 + 0.001\left(\bar{x}_1^2 + \bar{x}_2^2\right)}, \text{ где} \tag{2.26}$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = -10$, $Right_j = 10$, $j = \overline{1, n}$, $n = 2$.

Обозначение: $\bar{x}$ — вещественный вектор;

$n = 2$ — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{max} = \arg\max_{\bar{x} \in X} f\left(\bar{x}\right)$.

Точка максимума: $\bar{x}_{max} = (0, 0)^{\mathrm{T}}$, то есть $\left(\bar{x}_{max}\right)_j = 0$ $(j = \overline{1, n})$.

Максимум функции: $f\left(\bar{x}_{max}\right) = 1$.

График: Рисунок 2.26 нас 93 стр.

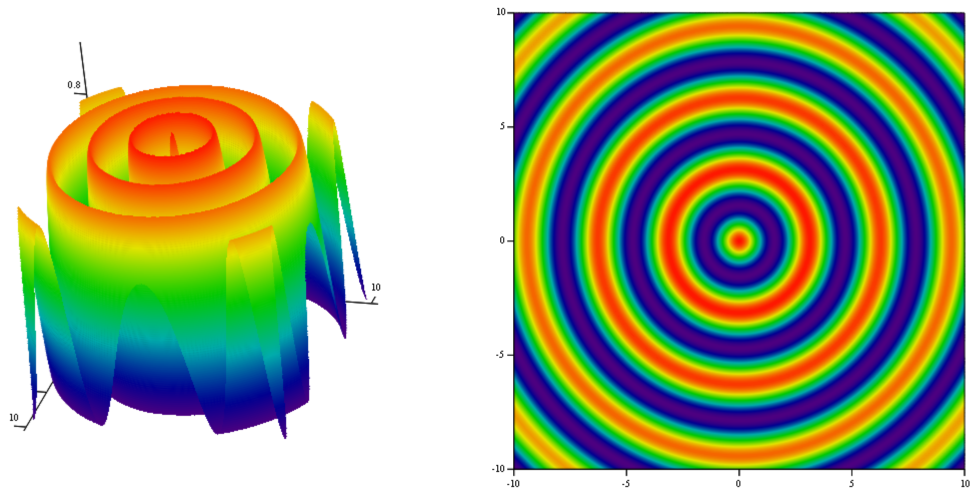


Рисунок 2.26. Функция Сомбреро

2.25.2 Параметры для алгоритмов оптимизации

Точность вычислений:	$\varepsilon = 0.05$.
Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) :	$NumberOfParts_j = 4095$ $(j = \overline{1, n})$.
Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) :	$(k_2)_j = 12$ $(j = \overline{1, n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } \left(k_2\right)_j \in \mathbb{N}, \left(j = \overline{1, n}\right).$$

2.25.3 Основная задача и подзадачи

Изменяемый параметр:	n — размерность вещественного вектора.
Значение в основной задаче:	$n = 2$.

2.25.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно $(k = \overline{1, N})$. Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1, n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N} \left(\frac{\sqrt{\sum_{j=1}^{n} \left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j \right)^2}}{n} \right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N} \left| f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right) \right|}{N}.$$

2.25.5 Свойства задачи

Условной или безусловной оптимизации:	Задача безусловной оптимизации.
Одномерной или многомерной оптимизации:	Многомерной: (двумерной).
Функция унимодальная или многоэкстремальная:	Функция многоэкстремальная.
Функция стохастическая или нет:	Функция не стохастическая.
Особенности:	Нет.

2.25.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.25. Код функции MHL_TestFunction_Sombrero

```
double MHL_TestFunction_Sombrero(double x, double y)
{
/*
Функция одной переменных: функция Сомбреро.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - первая вещественная переменная;
 y - вторая вещественная переменная.
Возвращаемое значение:
 Значение тестовой функции в точке (x,y).
*/
double VMHL_Result;
VMHL_Result = 1.-sin(sqrt(x*x+y*y))*sin(sqrt(x*x+y*y));
VMHL_Result /= (1.+0.001*(x*x+y*y));
return VMHL_Result;
}
```

2.25.7 Ссылки

Данная функция приводится в следующих источниках:

1. [18, стр. 30] — Эволюционные методы моделирования и оптимизации сложных систем.

2.26 Функция Multiextremal

2.26.1 Описание функции

Идентификатор: MHL_TestFunction_Multiextremal.

Наименование: Multiextremal.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = 0.05(x-1)^2 + \left(3 - 2.9e^{-2.77257x^2}\right)\left(1 - \cos\left(x\left(4 - 50e^{-2.77257x^2}\right)\right)\right), \text{ где} \quad (2.27)$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = -2$, $Right_j = 2$, $j = \overline{1,n}$, $n = 1$.

Обозначение: $\bar{x}$ — вещественный вектор;

$n = 1$ — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{min} = \arg\min_{\bar{x} \in X} f(\bar{x})$.

Точка минимума: $\bar{x}_{min} \approx (0.954452)^{\mathrm{T}}$, то есть $(\bar{x}_{min})_j \approx 0.954452$ $(j = \overline{1,n})$.

Минимум функции: $f(\bar{x}_{min}) \approx 0.000103742$.

График: Рисунок 2.27 на с 96 стр.

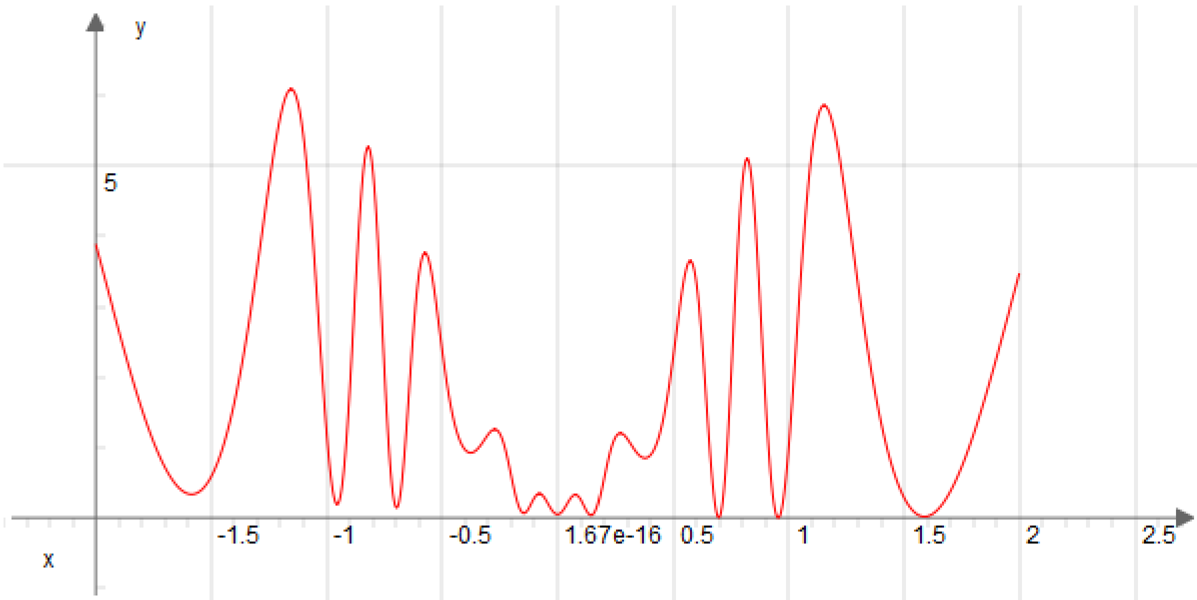


Рисунок 2.27. Функция Multiextremal

2.26.2 Параметры для алгоритмов оптимизации

Точность вычислений: $\varepsilon = 0.01$.

Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) : $NumberOfParts_j = 4095$ $(j = \overline{1, n})$.

Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) : $(k_2)_j = 12$ $(j = \overline{1, n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\,(Right_j - Left_j)}{\varepsilon}, \text{где } (k_2)_j \in \mathbb{N}, \left(j = \overline{1, n}\right).$$

2.26.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 1$.

2.26.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно ($k = \overline{1, N}$). Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1, n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N}\left(\frac{\sqrt{\sum_{j=1}^{n}\left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n}\right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N}\left|f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.26.5 Свойства задачи

Условной или безусловной оптимизации: Задача безусловной оптимизации.

Одномерной или многомерной оптимизации: Одномерной.

Функция унимодальная или многоэкстремальная: Функция многоэкстремальная.

Функция стохастическая или нет: Функция не стохастическая.

Особенности: Нет.

2.26.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.26. Код функции MHL_TestFunction_Multiextremal

```
double MHL_TestFunction_Multiextremal(double x)
{
/*
Функция одной переменных: функция Multiextremal.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - вещественная переменная.
Возвращаемое значение:
 Значение тестовой функции в точке (x).
*/
double VMHL_Result;
VMHL_Result = (0.05*(x-1.)*(x-1.)+(3.-2.9*exp(-2.77257*x*x))*(1-cos(x*(4.-50*exp
   (-2.77257*x*x)))));
return VMHL_Result;
}
```

2.26.7 Ссылки

Данная функция приводится в следующих источниках:

1. [18, стр. 26] — Эволюционные методы моделирования и оптимизации сложных систем.

По сравнению с данной работой в данном документе представлено уточненное значение функции в точке минимума.

2.27 Функция Multiextremal2

2.27.1 Описание функции

Идентификатор: MHL_TestFunction_Multiextremal2.

Наименование: Multiextremal2.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = 1 - 0.5\cos\left(1.5\left(10x - 0.3\right)\right)\cos\left(31.4x\right) + 0.5\cos\left(\sqrt{5}\cdot 10x\right)\cos\left(35x\right), \text{ где} \quad (2.28)$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = -2$, $Right_j = 2$, $j = \overline{1, n}$, $n = 1$.

Обозначение: $\bar{x}$ — вещественный вектор;

$n = 1$ — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{max} = \arg\max_{\bar{x}\in X} f(\bar{x})$.

Точка максимума: $\bar{x}_{max} \approx (-0.993263)^{\mathrm{T}}$, то есть $(\bar{x}_{max})_j \approx -0.993263$ $(j = \overline{1, n})$.

Максимум функции: $f(\bar{x}_{max}) \approx 1.93374$.

График: Рисунок 2.28 нас 99 стр.

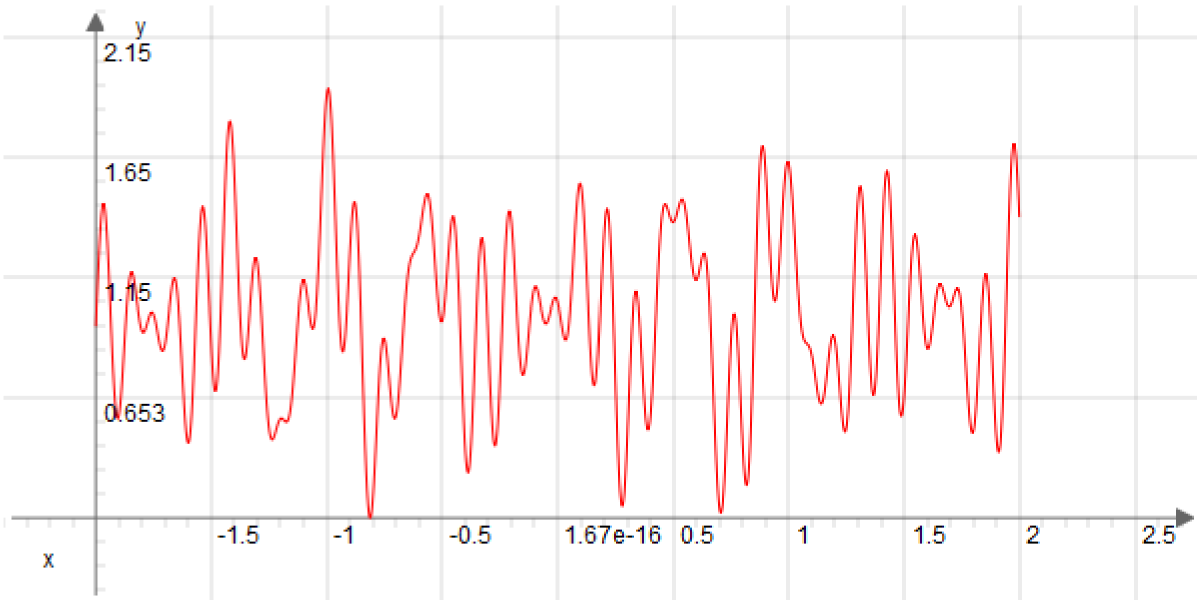


Рисунок 2.28. Функция Multiextremal2

2.27.2 Параметры для алгоритмов оптимизации

Точность вычислений:	$\varepsilon = 0.01$.
Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) :	$NumberOfParts_j = 4095$ $(j = \overline{1,n})$.
Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) :	$(k_2)_j = 12$ $(j = \overline{1,n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\left(Right_j - Left_j\right)}{\varepsilon}, \text{где } \left(k_2\right)_j \in \mathbb{N}, \left(j = \overline{1,n}\right).$$

2.27.3 Основная задача и подзадачи

Изменяемый параметр:	n — размерность вещественного вектора.
Значение в основной задаче:	$n = 1$.

2.27.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно $(k = \overline{1,N})$. Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1,n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N}\left(\frac{\sqrt{\sum_{j=1}^{n}\left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right)^2}}{n}\right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N}\left|f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right)\right|}{N}.$$

2.27.5 Свойства задачи

Условной или безусловной оптимизации: Задача безусловной оптимизации.

Одномерной или многомерной оптимизации: Одномерной.

Функция унимодальная или многоэкстремальная: Функция многоэкстремальная.

Функция стохастическая или нет: Функция не стохастическая.

Особенности: Нет.

2.27.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.27. Код функции MHL_TestFunction_Multiextremal2

```
double MHL_TestFunction_Multiextremal2(double x)
{
/*
Функция одной переменных: функция Multiextremal2.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - вещественная переменная.
Возвращаемое значение:
 Значение тестовой функции в точке (x).
*/
double VMHL_Result;
VMHL_Result = 1.-0.5*cos(1.5*(10.*x-0.3))*cos(31.4*x)+0.5*cos(sqrt(5.)*10.*x)*cos
    (35.*x);
return VMHL_Result;
}
```

2.27.7 Ссылки

Данная функция приводится в следующих источниках:

1. [18, стр. 27] — Эволюционные методы моделирования и оптимизации сложных систем.

По сравнению с данной работой в данном документе представлено уточненное значение функции в точке минимума.

2.28 Функция волна

2.28.1 Описание функции

Идентификатор: MHL_TestFunction_Wave.

Наименование: Волна.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = e^{-\bar{x}_1^2} + 0.01\cos(200 \cdot \bar{x}_1), \text{ где} \quad (2.29)$$

$\bar{x} \in X$, $\bar{x}_j \in [Left_j; Right_j]$, $Left_j = -2$, $Right_j = 2$, $j = \overline{1, n}$, $n = 1$.

Обозначение: $\bar{x}$ — вещественный вектор;

$n = 1$ — размерность вещественного вектора.

Решаемая задача оптимизации: $\bar{x}_{max} = \arg\max_{\bar{x} \in X} f(\bar{x})$.

Точка максимума: $\bar{x}_{max} = (0)^{\mathrm{T}}$, то есть $(\bar{x}_{max})_j = 0$ $(j = \overline{1, n})$.

Максимум функции: $f(\bar{x}_{max}) = 1.01$.

График: Рисунок 2.29 нас 102 стр.

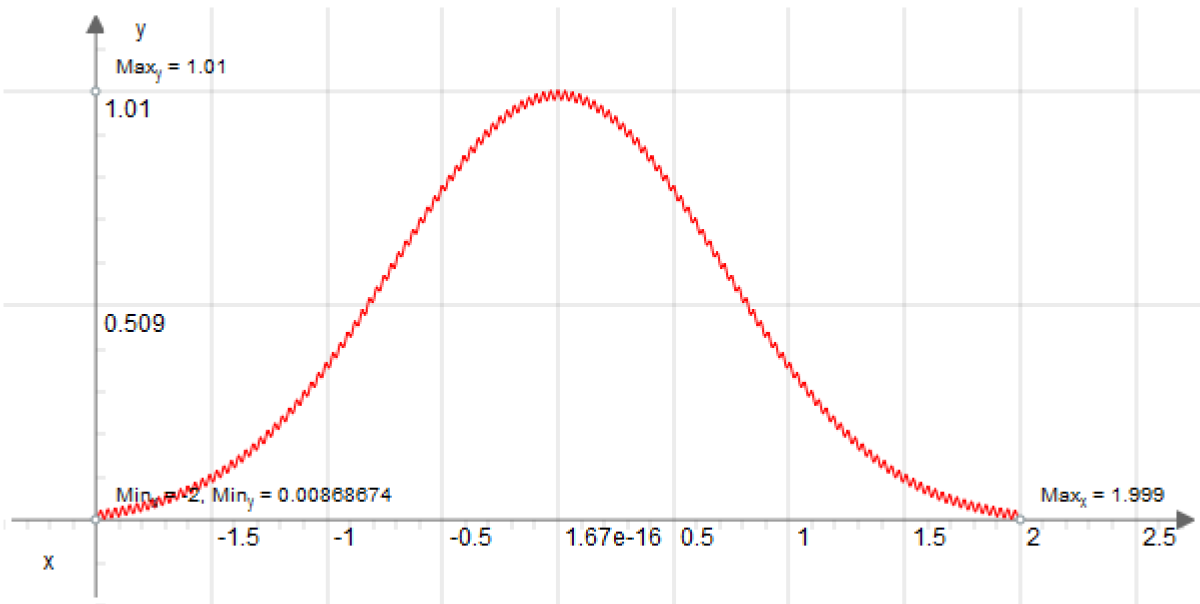


Рисунок 2.29. Волна

2.28.2 Параметры для алгоритмов оптимизации

Точность вычислений: $\varepsilon = 0.01$.

Число интервалов, на которые предполагается разбивать каждую компоненту вектора $\bar{x}$ в пределах своего изменения (для алгоритмов дискретной оптимизации) : $NumberOfParts_j = 4095$ $(j = \overline{1, n})$.

Для этого длина бинарной строки для x_j координаты равна (для алгоритмов бинарной оптимизации) : $(k_2)_j = 12$ $(j = \overline{1, n})$.

Замечание: $NumberOfParts_j$ выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10\,(Right_j - Left_j)}{\varepsilon}, \text{где } (k_2)_j \in \mathbb{N}, \left(j = \overline{1, n}\right).$$

2.28.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 1$.

2.28.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submin}$ со значениями целевой функции $f\left(\bar{x}^k_{submin}\right)$ соответственно $(k = \overline{1, N})$. Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submin}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submin}\right) = \begin{cases} 1, \text{ если } \left|\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j\right| < \varepsilon, j = \overline{1, n}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N} \left(\frac{\sqrt{\sum_{j=1}^{n} \left(\left(\bar{x}^k_{submin}\right)_j - \left(\bar{x}_{min}\right)_j \right)^2}}{n} \right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N} \left| f\left(\bar{x}^k_{submin}\right) - f\left(\bar{x}_{min}\right) \right|}{N}.$$

2.28.5 Свойства задачи

Условной или безусловной оптимизации:	Задача безусловной оптимизации.
Одномерной или многомерной оптимизации:	Одномерной.
Функция унимодальная или многоэкстремальная:	Функция многоэкстремальная.
Функция стохастическая или нет:	Функция не стохастическая.
Особенности:	Хотя внешне можно отнести эту функцию к стохастической, так как по поведению напоминает вид плотности нормального распределения с помехой.

2.28.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 2.28. Код функции MHL_TestFunction_Wave

```
double MHL_TestFunction_Wave(double x)
{
/*
Функция одной переменных: волна.
Тестовая функция вещественной оптимизации.
Входные параметры:
 x - вещественная переменная.
Возвращаемое значение:
 Значение тестовой функции в точке (x).
*/
double VMHL_Result;
VMHL_Result = (exp(-x*x)+0.01*cos(200*x));
return VMHL_Result;
}
```

2.28.7 Ссылки

Так и не смог найти нормальный источник для этой функции. Откуда-то у меня находится со студенческих времен.

Глава 3

Задачи бинарной оптимизации

3.1 Сумма всех элементов бинарного вектора

3.1.1 Описание функции

Идентификатор: MHL_TestFunction_SumVector.

Наименование: Сумма всех элементов бинарного вектора.

Тип: Задача бинарной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right) = \sum_{i=1}^{n} \bar{x}_i, \text{ где} \tag{3.1}$$

$\bar{x} \in X$, $\bar{x}_j \in \{0; 1\}$, $j = \overline{1, n}$.

Обозначение: $\bar{x}$ — бинарный вектор;

n — размерность бинарного вектора.

Объем поискового пространства: $\mu\left(X\right) = 2^n$.

Решаемая задача оптимизации: $\bar{x}_{max} = \arg\max_{\bar{x} \in X} f\left(\bar{x}\right)$.

Точка максимума: $\bar{x}_{max} = \left(1, 1, \ldots, 1\right)^{\mathrm{T}}$, то есть $\left(\bar{x}_{max}\right)_j = 1$ $(j = \overline{1, n})$.

Максимум функции: $f\left(\bar{x}_{max}\right) = n$.

Точка минимума: $\bar{x}_{min} = \left(0, 0, \ldots, 0\right)^{\mathrm{T}}$, то есть $\left(\bar{x}_{min}\right)_j = 0$ $(j = \overline{1, n})$.

Минимум функции: $f\left(\bar{x}_{min}\right) = 0$.

3.1.2 Основная задача и подзадачи

Изменяемый параметр:	n — размерность бинарного вектора.
Значение в основной задаче:	$n = 20$.
Подзадача №2:	$n = 30$.
Подзадача №3:	$n = 40$.
Подзадача №4:	$n = 50$.
Подзадача №5:	$n = 60$.
Подзадача №6:	$n = 70$.
Подзадача №7:	$n = 80$.
Подзадача №8:	$n = 90$.
Подзадача №9:	$n = 100$.
Подзадача №10:	$n = 200$.

3.1.3 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за N запусков мы нашли решения $\bar{x}^k_{submax}$ со значениями целевой функции $f\left(\bar{x}^k_{submax}\right)$ соответственно ($k = \overline{1, N}$). Используем три вида ошибок:

Надёжность:

$$R = \frac{\sum_{k=1}^{N} S\left(\bar{x}^k_{submax}\right)}{N}, \text{ где}$$

$$S\left(\bar{x}^k_{submax}\right) = \begin{cases} 1, \text{ если } \bar{x}^k_{submax} = \bar{x}_{max}; \\ 0, \text{ иначе.} \end{cases}$$

Ошибка по входным параметрам:

$$E_x = \frac{\sum_{k=1}^{N} \left(\frac{\sum_{j=1}^{n} \left| \left(\bar{x}^k_{submax}\right)_j - \left(\bar{x}_{max}\right)_j \right|}{n} \right)}{N}.$$

Ошибка по значениям целевой функции:

$$E_f = \frac{\sum_{k=1}^{N} \left(\frac{\left| f\left(\bar{x}^k_{submax}\right) - f\left(\bar{x}_{max}\right) \right|}{n} \right)}{N}.$$

3.1.4 Свойства задачи

Условной или безусловной оптимизации: Задача безусловной оптимизации.

Одномерной или многомерной оптимизации: Многомерной: n.

Функция унимодальная или многоэкстремальная: Функция унимодальная.

Функция стохастическая или нет: Функция не стохастическая.

Особенности: Нет.

3.1.5 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу https://github.com/Harrix/HarrixMathLibrary.

Код 3.1. Код функции MHL_TestFunction_SumVector

```
double MHL_TestFunction_SumVector(int *x, int VMHL_N)
{
/*
Сумма всех элементов бинарного вектора.
Тестовая функция бинарной оптимизации.
Входные параметры:
 x - указатель на исходный массив;
 VMHL_N - размер массива x.
Возвращаемое значение:
 Значение тестовой функции в точке x.
*/
double VMHL_Result=0;
for (int i=0;i<VMHL_N;i++) VMHL_Result+=x[i];
return VMHL_Result;
}
```

Литература

1. Dieterich Johannes M., Hartke Bernd. Empirical review of standard benchmark functions using evolutionary global optimization // CoRR. 2012. T. abs/1207.4318. http://arxiv.org/pdf/1207.4318v1.pdf.

2. Ackley's Function. http://www.cs.unm.edu/~neal.holts/dga/benchmarkFunction/ackley.html.

3. Yuret D. De Jong's test suite. 1997. http://www2.denizyuret.com/pub/aitr1569/node19.html.

4. Jamil M., Yang X.-S. A Literature Survey of Benchmark Functions For Global Optimization Problems // ArXiv e-prints. 2013. aug. http://arxiv.org/pdf/1308.4008.pdf.

5. Pohlheim Hartmut. GEATbx Examples. Examples of Objective Functions. 2006. http://www.geatbx.com/download/GEATbx_ObjFunExpl_v38.pdf.

6. Paraboloid. http://en.wikipedia.org/wiki/Paraboloid.

7. Rastrigin function. http://en.wikipedia.org/wiki/Rastrigin_function.

8. Non-linear Continuous Multi-Extremal Optimization. http://www.maths.uq.edu.au/CEToolBox/node3.html.

9. Parametric Optimization. http://www.pg.gda.pl/~mkwies/dyd/geadocu/fcnfun6.html.

10. Rosenbrock function. http://en.wikipedia.org/wiki/Rosenbrock_function.

11. Rosenbrock Function. http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page2537.htm.

12. Virtual Library of Simulation Experiments: Test Functions and Datasets. Rotated hyper-ellipsoid function. 2013. http://www.sfu.ca/~ssurjano/rothyp.html.

13. Benchmark Problems. http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume24/ortizboyer05a-html/node6.html.

14. Optimization Test Problems - Schwefel Function. http://www.sfu.ca/~ssurjano/schwef.html.

15. Test functions for Unconstrained Global Optimization - Schwefel Function. http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page2530.htm.

16. Parametric Optimization. Schwefel's function 7. http://www.pg.gda.pl/~mkwies/dyd/geadocu/fcnfun7.html.

17. Huang D.S., Li K., Irwin G.W. International Conference on Intelligent Computing: Intelligent computing. International Conference on Intelligent Computing: ICIC 2006, Kunming, China, August 16-19, 2006 : Proceedings. Springer, 2006. http://books.google.ru/books?id=7sH4RsXYu7cC.

18. Эволюционные методы моделирования и оптимизации сложных систем / Е. С. Семенкин, М. Н. Жукова, В. Г. Жуков [и др.]. Красноярск: Федеральное агентство по образованию, Сибирский федеральный университет, 2007. 310 с. http://files.lib.sfu-kras.ru/ebibl/umkd/22/u_lectures.pdf.

19. Himmelblau's function. http://en.wikipedia.org/wiki/Himmelblau's_function.

20. Minimization of the Himmelblau Function. http://pythonhosted.org/algopy/examples/minimization/himmelblau_minimization.html.

21. Rana. http://www.cs.unm.edu/~neal.holts/dga/benchmarkFunction/rana.html.