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.1.22

Красноярск – 2013

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 C-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 **1.22** от 29 декабря 2013 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

Задачи вещественной оптимизации

1.1 Функция Ackley

1.1.1 Описание функции

Идентификатор: MHL_TestFunction_Ackley.

Наименование: Функция Ackley.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

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

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

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

n — размерность вещественного вектора.

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

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 1.1 нас 11 стр.

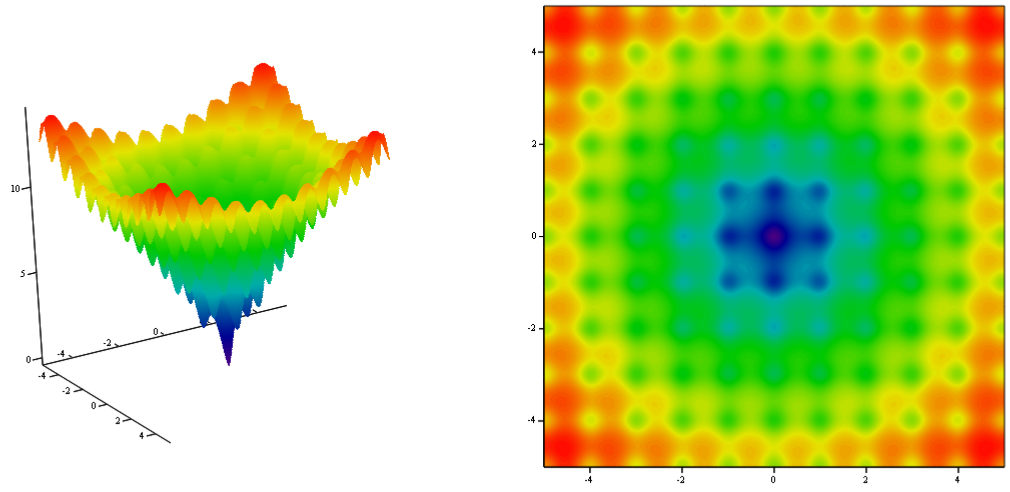

Рисунок 1.1. Функция Ackley

1.1.2 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:

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

1.1.3 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: $n = 30$.

1.1.4 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:

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

1.1.5 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.

1.1.6 Реализация

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

Код 1.1. 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;
}
```

1.1.7 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.

1.2 Функция Гипер-эллипсоид

1.2.1 Описание функции

Идентификатор: MHL_TestFunction_HyperEllipsoid.

Наименование: Гипер-эллипсоид.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right)=\sum_{i=1}^{n}\left(i\cdot\bar{x}_i\right)^2, \text{ где} \tag{1.2}$$

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

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 1.2 нас 14 стр.

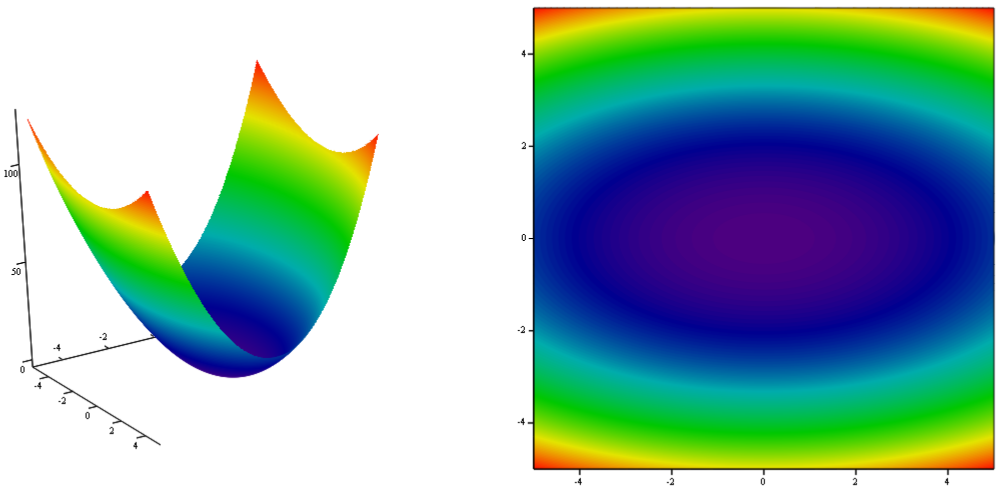

Рисунок 1.2. Гипер-эллипсоид

1.2.2 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) : $\left(k_2\right)_j = 12$ $(j = \overline{1, n})$.

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

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

1.2.3 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: $n = 30$.

1.2.4 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:

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

1.2.5 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 не стохастическая.

Особенности: Нет.

1.2.6 Реализация

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

Код 1.2. 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;
}
```

1.2.7 Ссылки

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

1.3 Эллиптический параболоид

1.3.1 Описание функции

Идентификатор: MHL_TestFunction_ParaboloidOfRevolution.

Наименование: Эллиптический параболоид.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = \sum_{i=1}^{n} \bar{x}_i^2, \text{ где} \tag{1.3}$$

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

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

n — размерность вещественного вектора.

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

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 1.3 на с 17 стр.

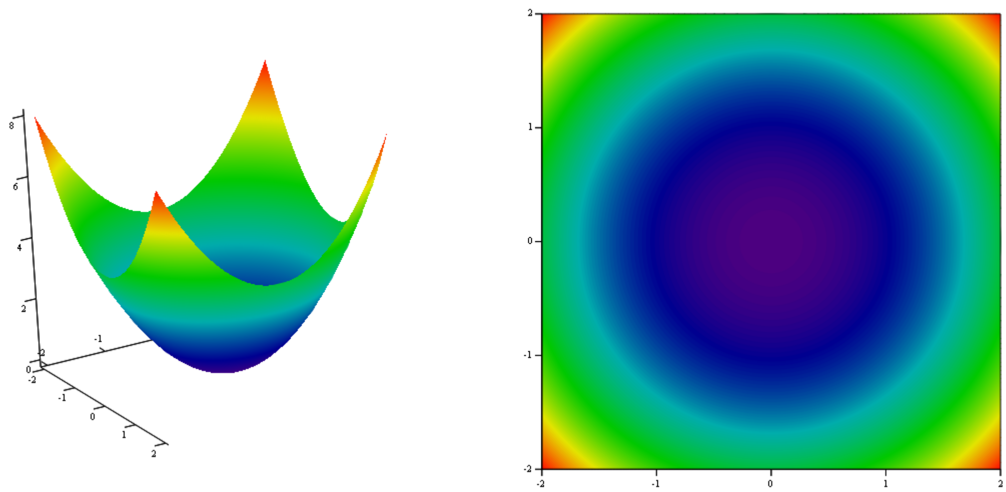

Рисунок 1.3. Эллиптический параболоид

1.3.2 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:

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

1.3.3 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:	$n = 30$.

1.3.4 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:

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

1.3.5 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.

1.3.6 Реализация

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

Код 1.3. 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;
}
```

1.3.7 Ссылки

Данная функция приводится в следующих источниках:

1. [3] — Paraboloid.

1.4 Функция Растригина

1.4.1 Описание функции

Идентификатор: MHL_TestFunction_Rastrigin.

Наименование: Функция Растригина.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right) = 10n + \sum_{i=1}^{n} \left(\bar{x}_i^2 - 10 \cdot \cos\left(2\pi \cdot \bar{x}_i\right)\right), \text{ где} \tag{1.4}$$

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

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

n — размерность вещественного вектора.

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

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 1.4 нас 20 стр.

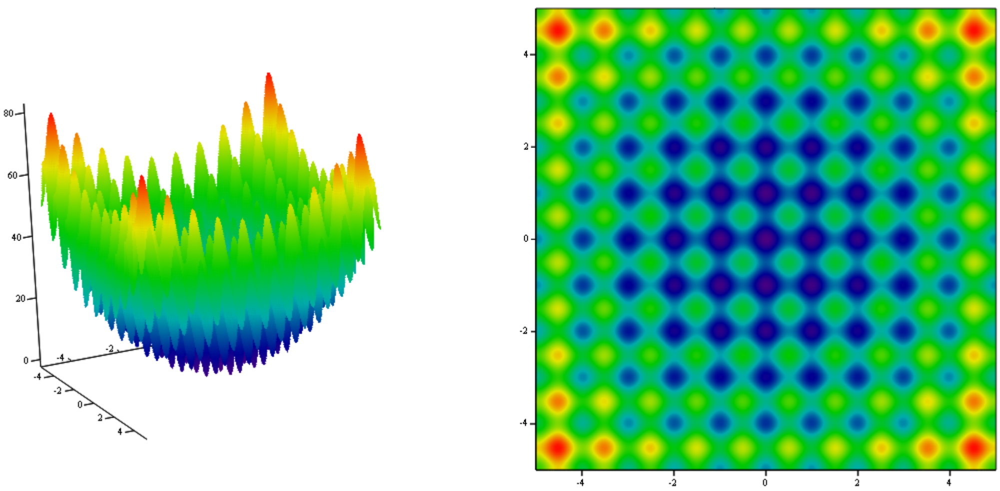

Рисунок 1.4. Функция Растригина

1.4.2 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:

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

1.4.3 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:	$n = 30.$

1.4.4 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:

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

Ошибка по значениям целевой функции:

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

1.4.5 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.

1.4.6 Реализация

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

Код 1.4. 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;
}
```

1.4.7 Ссылки

Данная функция приводится в следующих источниках:

1. [4] — Rastrigin function.
2. [5] — Non-linear Continuous Multi-Extremal Optimization.
3. [6] — Parametric Optimization.

1.5 Функция Розенброка

1.5.1 Описание функции

Идентификатор:	MHL_TestFunction_Rosenbrock.
Наименование:	Функция Розенброка.
Тип:	Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = \sum_{i=1}^{n-1} \left(100 \left(\bar{x}_{i+1} - \bar{x}_i^2 \right)^2 + (1 - \bar{x}_i)^2 \right), \text{ где} \tag{1.5}$$

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

Обозначение:	$\bar{x}$ — вещественный вектор;
	n — размерность вещественного вектора.
Решаемая задача оптимизации:	$\bar{x}_{min} = \arg\min_{\bar{x} \in X} f(\bar{x})$.
Точка минимума:	$\bar{x}_{min} = (1, 1, \ldots, 1)^{\mathrm{T}}$, то есть $(\bar{x}_{min})_j = 1$ $(j = \overline{1, n})$.
Минимум функции:	$f(\bar{x}_{min}) = 0$.
График:	Рисунок 1.5 нас 24 стр.

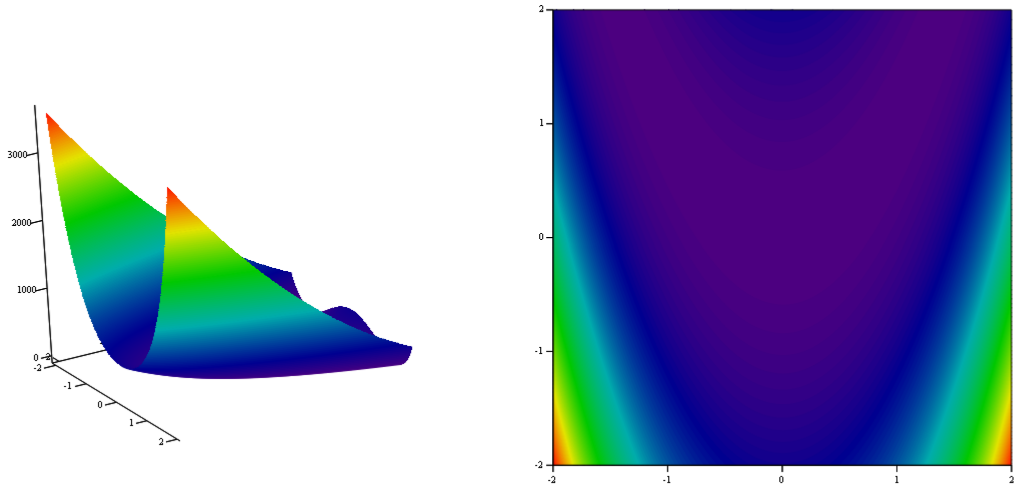

Рисунок 1.5. Функция Розенброка

1.5.2 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:

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

1.5.3 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: $n = 30.$

1.5.4 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:

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

1.5.5 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.

1.5.6 Реализация

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

Код 1.5. 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;
}
```

1.5.7 Ссылки

Данная функция приводится в следующих источниках:

1. [7] — Rosenbrock function.
2. [8] — Rosenbrock Function.

1.6 Функция Развернутый гипер-эллипсоид

1.6.1 Описание функции

Идентификатор: MHL_TestFunction_RotatedHyperEllipsoid.

Наименование: Развернутый гипер-эллипсоид.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{j} \bar{x}_j \right)^2, \text{ где} \tag{1.6}$$

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

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

n — размерность вещественного вектора.

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

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 1.6 нас 27 стр.

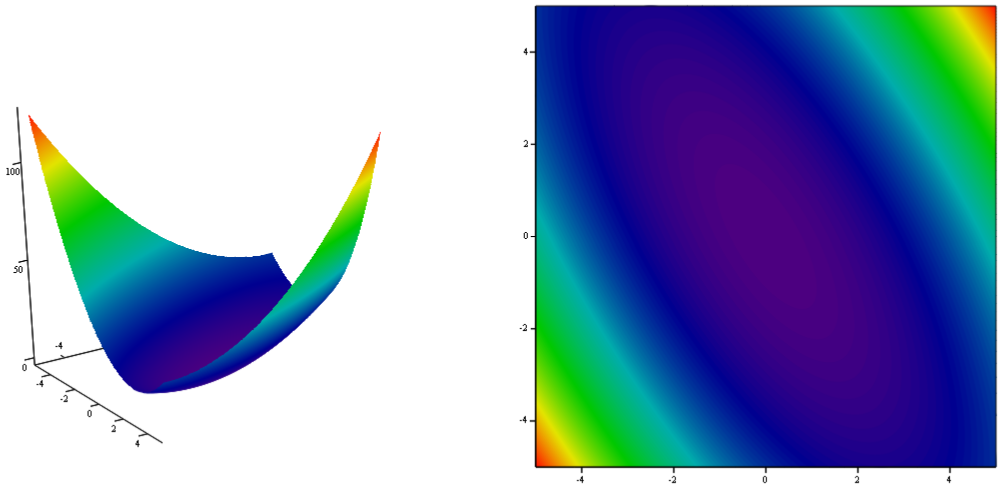

Рисунок 1.6. Развернутый гипер-эллипсоид

1.6.2 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:

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

1.6.3 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: $n = 30$.

1.6.4 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:

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

1.6.5 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.

1.6.6 Реализация

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

Код 1.6. 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;
}
```

1.6.7 Ссылки

Данная функция приводится в следующих источниках:

1. [9, стр. 4] — GEATbx Examples. Examples of Objective Functions.

Обратите внимание, что иногда под названием Rotated Hyper Ellipsoid встречается (например, [10]) 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.

1.7 Функция Step (модифицированная версия De Jong 3)

1.7.1 Описание функции

Идентификатор:	MHL_TestFunction_StepFunction.
Наименование:	Функция Step (модифицированная версия De Jong 3).
Тип:	Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = \begin{cases} \sum\limits_{i=1}^{n} \left(int\left(\bar{x}_i\right)\right)^2, \text{ если } \sum\limits_{i=1}^{n} \left|int\left(\bar{x}_i\right)\right| \neq 0; \\ \left(\sum\limits_{i=1}^{n} \left|\bar{x}_i\right|\right) - 1, \text{ иначе.} \end{cases}, \text{ где} \tag{1.7}$$

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

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 1.7 нас 30 стр., 1.8 нас 30 стр.

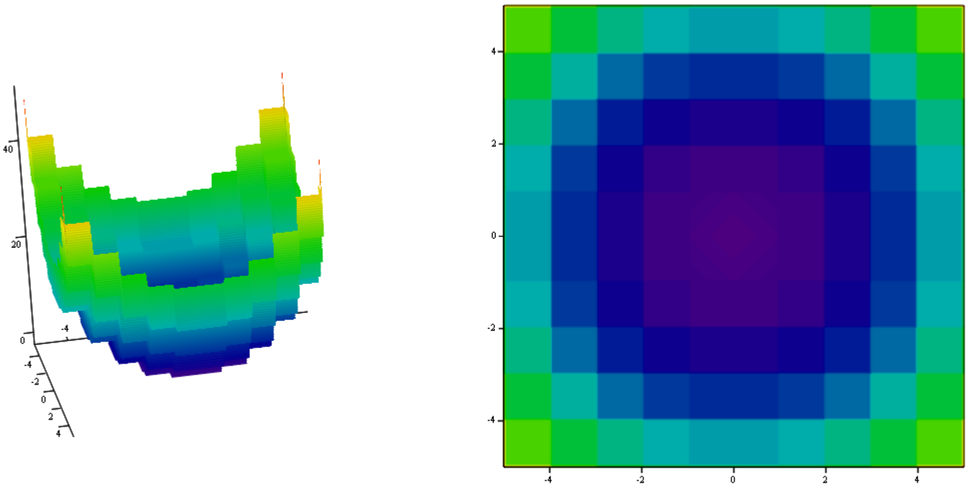

Рисунок 1.7. Функция Step (модифицированная версия De Jong 3)

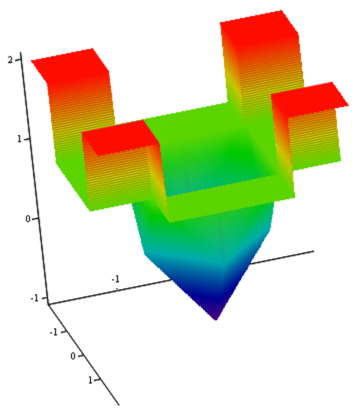

Рисунок 1.8. Функция Step (модифицированная версия De Jong 3) в области около точки минимума

1.7.2 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:

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

1.7.3 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:	$n = 30$.

1.7.4 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:

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

Ошибка по значениям целевой функции:

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

1.7.5 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.

1.7.6 Реализация

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

Код 1.7. 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;
}
```

1.7.7 Ссылки

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

1. [11, стр. 729] — International Conference on Intelligent Computing: Intelligent computing.

1.8 Аддитивная потенциальная функция

1.8.1 Описание функции

Идентификатор: MHL_TestFunction_AdditivePotential.

Наименование: Аддитивная потенциальная функция.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right) = z\left(\bar{x}_1\right) + z\left(\bar{x}_2\right), \text{ где} \tag{1.8}$$

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

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

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

Минимум функции: $f\left(\bar{x}_{min}\right) = -15.6060606060606$.

График: Рисунок 1.9 нас 34 стр.

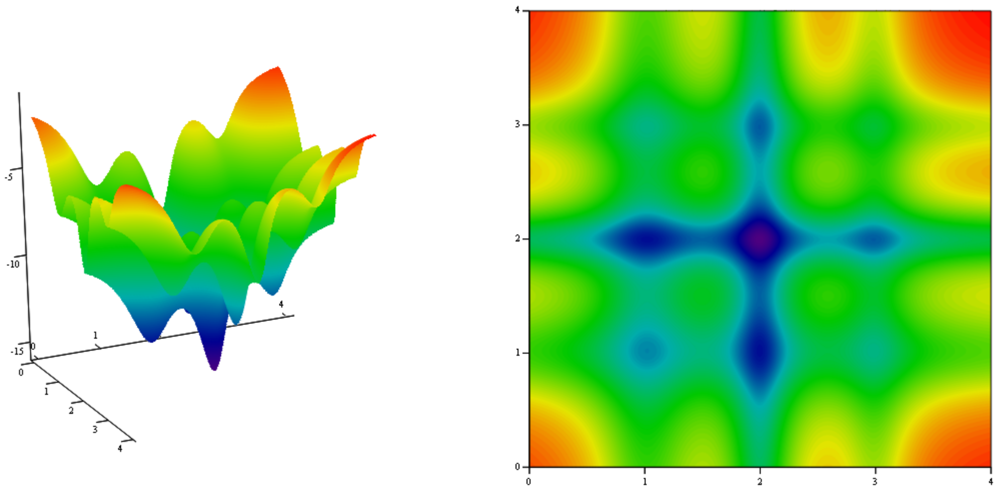

Рисунок 1.9. Аддитивная потенциальная функция

1.8.2 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:

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

1.8.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

1.8.4 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:

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

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

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:

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

1.8.5 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.

1.8.6 Реализация

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

Код 1.8. 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;
}
```

1.8.7 Ссылки

Данная функция приводится в следующих источниках:

1. [12, стр. 33] — Эволюционные методы моделирования и оптимизации сложных систем.

1.9 Функция Egg Holder

1.9.1 Описание функции

Идентификатор: MHL_TestFunction_EggHolder.

Наименование: Функция Egg Holder.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right) = -\bar{x}_1 \sin\left(\sqrt{\left|\bar{x}_1 - 47 - \bar{x}_2\right|}\right) - \left(\bar{x}_2 + 47\right) \sin\left(\sqrt{\left|\frac{\bar{x}_1}{2} + 47 + \bar{x}_2\right|}\right), \text{ где} \quad (1.9)$$

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

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 1.10 на с 37 стр.

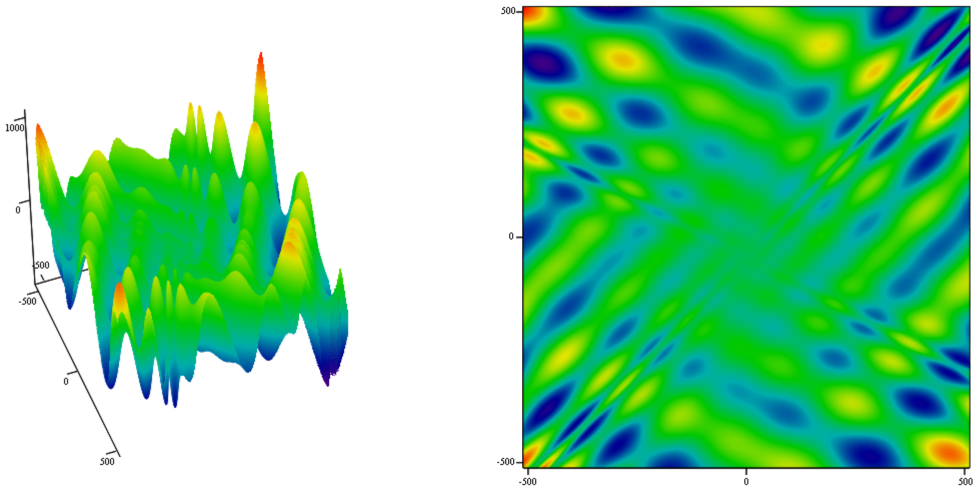

Рисунок 1.10. Функция Egg Holder

1.9.2 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:

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

1.9.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2.$

1.9.4 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:

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

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

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:

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

1.9.5 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.

1.9.6 Реализация

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

Код 1.9. 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;
}
```

1.9.7 Ссылки

Данная функция приводится в следующих источниках:

1. [13, стр. 15] — A Literature Survey of Benchmark Functions For Global Optimization Problems.

1.10 Функция Химмельблау

1.10.1 Описание функции

Идентификатор: MHL_TestFunction_Himmelblau.

Наименование: Функция Химмельблау.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = \left(\bar{x}_1^2 + \bar{x}_2 - 11\right)^2 + \left(\bar{x}_1 + \bar{x}^2 - 7\right)^2, \text{ где} \tag{1.10}$$

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

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

$n = 2$ — размерность вещественного вектора.

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

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

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

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

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

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

График: Рисунок 1.11 нас 40 стр.

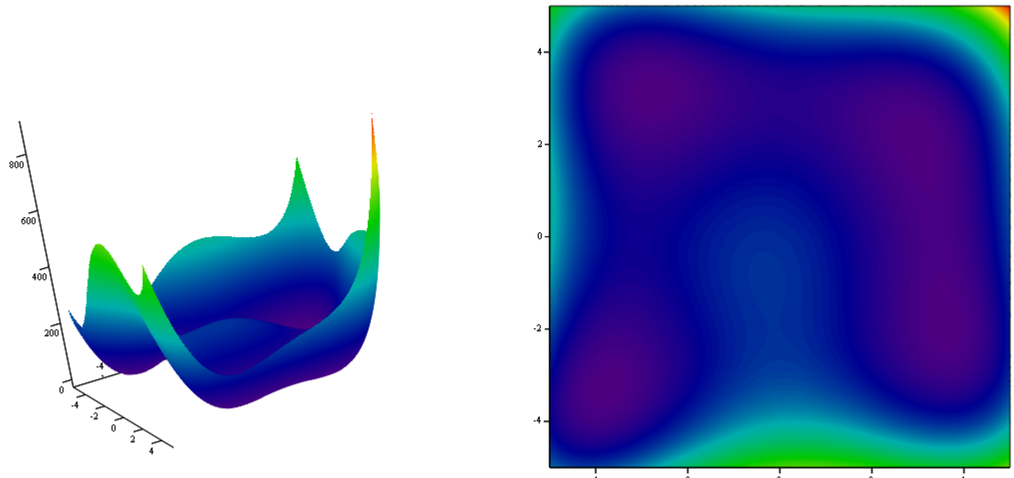

Рисунок 1.11. Функция Химмельблау

1.10.2 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:

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

1.10.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

1.10.4 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:

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

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

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)

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

1.10.5 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.

1.10.6 Реализация

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

Код 1.10. 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;
}
```

1.10.7 Ссылки

Данная функция приводится в следующих источниках:

1. [14] — Himmelblau's function.
2. [15] — Minimization of the Himmelblau Function.

1.11 Функция Катникова

1.11.1 Описание функции

Идентификатор: MHL_TestFunction_Katnikov.

Наименование: Функция Катникова.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = 0.5\left(\bar{x}_1^2+\bar{x}_2^2\right)\left(2A + A\cos\left(1.5\bar{x}_1\right)\cos\left(3.14\bar{x}_2\right) + A\cos\left(\sqrt{5}\bar{x}_1\right)\cos\left(3.5\bar{x}_2\right)\right), \text{ где} \quad (1.11)$$

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

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 1.12 нас 43 стр.

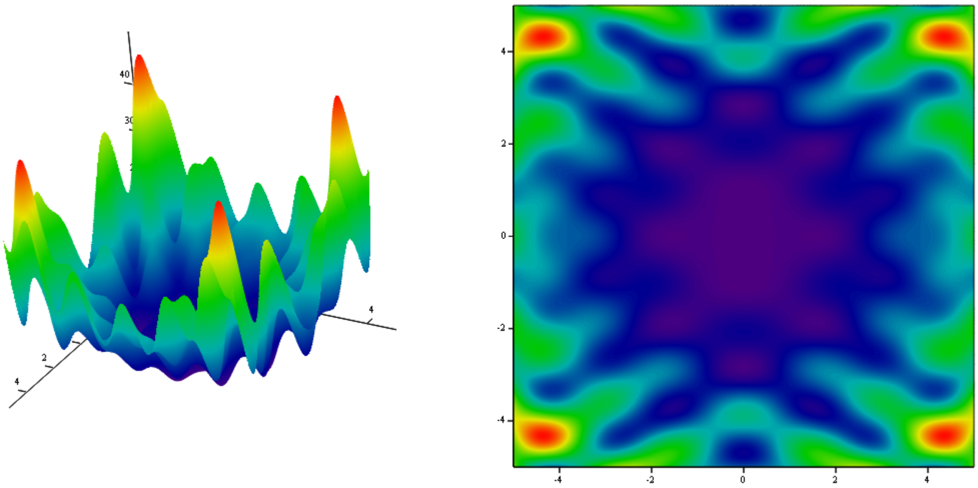

Рисунок 1.12. Функция Катникова

1.11.2 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:

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

1.11.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

1.11.4 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:

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

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

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:

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

1.11.5 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.

1.11.6 Реализация

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

Код 1.11. 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;
}
```

1.11.7 Ссылки

Данная функция приводится в следующих источниках:

1. [12, стр. 31] — Эволюционные методы моделирования и оптимизации сложных систем.

1.12 Функция Multiextremal3

1.12.1 Описание функции

Идентификатор: MHL_TestFunction_Multiextremal3.

Наименование: Функция Multiextremal3.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = \bar{x}_1^2 \left|\sin(2\bar{x}_1)\right| + \bar{x}_2^2 \left|\sin(2\bar{x}_2)\right| - \frac{1}{5\bar{x}_1^2 + 5\bar{x}_2^2 + 0.2} + 5, \text{ где} \tag{1.12}$$

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

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 1.13 нас 46 стр.

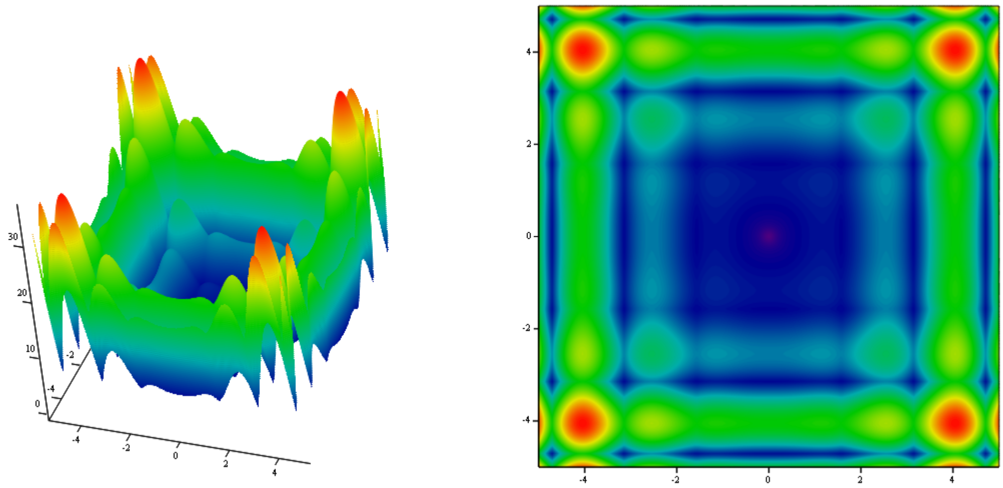

Рисунок 1.13. Функция Multiextremal3

1.12.2 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:

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

1.12.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

1.12.4 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:

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

1.12.5 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.

1.12.6 Реализация

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

Код 1.12. 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;
}
```

1.12.7 Ссылки

Данная функция приводится в следующих источниках:

1. [12, стр. 31] — Эволюционные методы моделирования и оптимизации сложных систем.

1.13 Функция Multiextremal4

1.13.1 Описание функции

Идентификатор: MHL_TestFunction_Multiextremal4.

Наименование: Функция Multiextremal4.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = 0.5\left(\bar{x}_1^2 + \bar{x}_1\bar{x}_2 + \bar{x}_2^2\right)\left(1 + 0.5\cos(1.5\bar{x}_1)\cos(3.2\bar{x}_1\bar{x}_2)\cos(3.14\bar{x}_2) + \right.$$
$$\left. +0.5\cos(2.2\bar{x}_1)\cos(4.8\bar{x}_1\bar{x}_2)\cos(3.5\bar{x}_2)\right), \text{ где} \quad (1.13)$$

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

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

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

График: Рисунок 1.14 нас 49 стр.

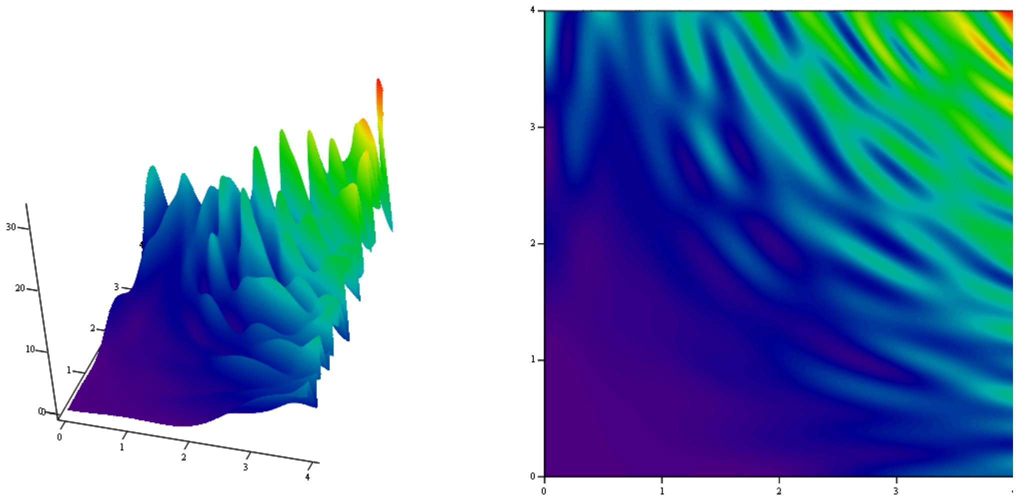

Рисунок 1.14. Функция Multiextremal4

1.13.2 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:

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

1.13.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

1.13.4 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:

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

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

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:

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

1.13.5 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.

1.13.6 Реализация

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

Код 1.13. 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;
}
```

1.13.7 Ссылки

Данная функция приводится в следующих источниках:

1. [12, стр. 31] — Эволюционные методы моделирования и оптимизации сложных систем.

1.14 Мультипликативная потенциальная функция

1.14.1 Описание функции

Идентификатор: MHL_TestFunction_MultiplicativePotential.

Наименование: Мультипликативная потенциальная функция.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right)=-z\left(\bar{x}_1\right)\cdot z\left(\bar{x}_2\right),\text{ где} \tag{1.14}$$

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

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

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

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

График: Рисунок 1.15 на с 52 стр.

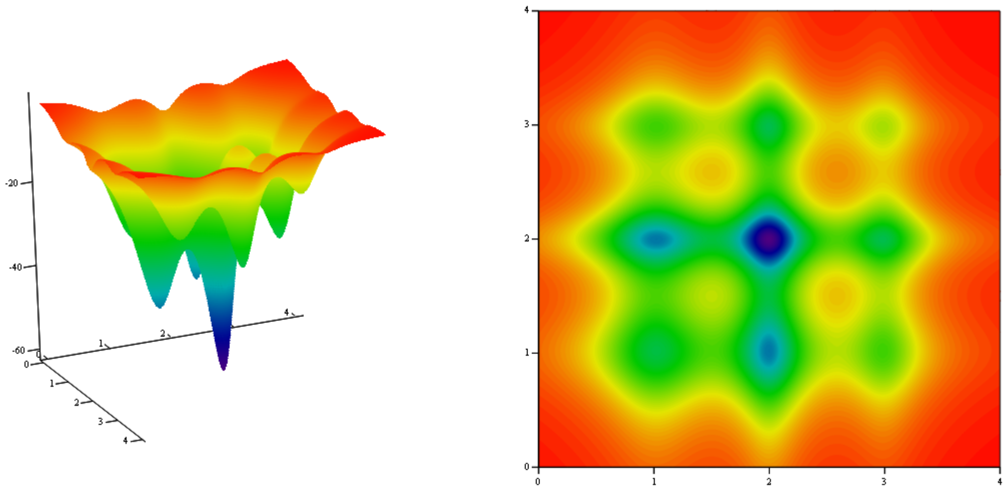

Рисунок 1.15. Мультипликативная потенциальная функция

1.14.2 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:

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

1.14.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

1.14.4 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:

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

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

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:

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

1.14.5 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:	Нет.

1.14.6 Реализация

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

Код 1.14. 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;
}
```

1.14.7 Ссылки

Данная функция приводится в следующих источниках:

1. [12, стр. 32] — Эволюционные методы моделирования и оптимизации сложных систем.

1.15 Функция Rana

1.15.1 Описание функции

Идентификатор: MHL_TestFunction_Rana.

Наименование: Функция Rana.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = \bar{x}_1 \sin\left(\sqrt{|\bar{x}_2 + 1 - \bar{x}_1|}\right) \cos\left(\sqrt{|\bar{x}_2 + 1 + \bar{x}_1|}\right) + \tag{1.15}$$

$$+ (\bar{x}_2 + 1) \cos\left(\sqrt{|\bar{x}_2 + 1 - \bar{x}_1|}\right) \sin\left(\sqrt{|\bar{x}_2 + 1 + x|}\right), \text{ где} \tag{1.16}$$

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

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

$n = 2$ — размерность вещественного вектора.

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

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

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

График: Рисунок 1.16 нас 55 стр.

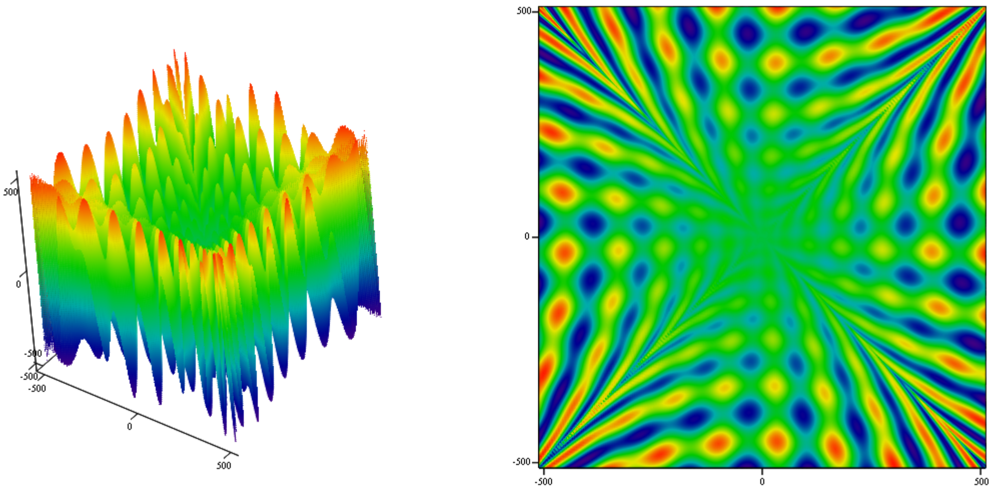

Рисунок 1.16. Функция Rana

1.15.2 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:

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

1.15.3 Основная задача и подзадачи

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

1.15.4 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:

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

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

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:

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

1.15.5 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.

1.15.6 Реализация

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

Код 1.15. 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;
}
```

1.15.7 Ссылки

Данная функция приводится в следующих источниках:

1. [16] — Rana.
2. [5] — Non-linear Continuous Multi-Extremal Optimization.

1.16 Функция ReverseGriewank

1.16.1 Описание функции

Идентификатор: MHL_TestFunction_ReverseGriewank.

Наименование: Функция ReverseGriewank.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right)=\frac{1}{\dfrac{\bar{x}_1^2+\bar{x}_2^2}{200}-cos\left(\bar{x}_0\right)cos\left(\dfrac{\bar{x}_2}{\sqrt{2}}\right)+2},\text{ где}\qquad(1.17)$$

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

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 1.17 нас 58 стр.

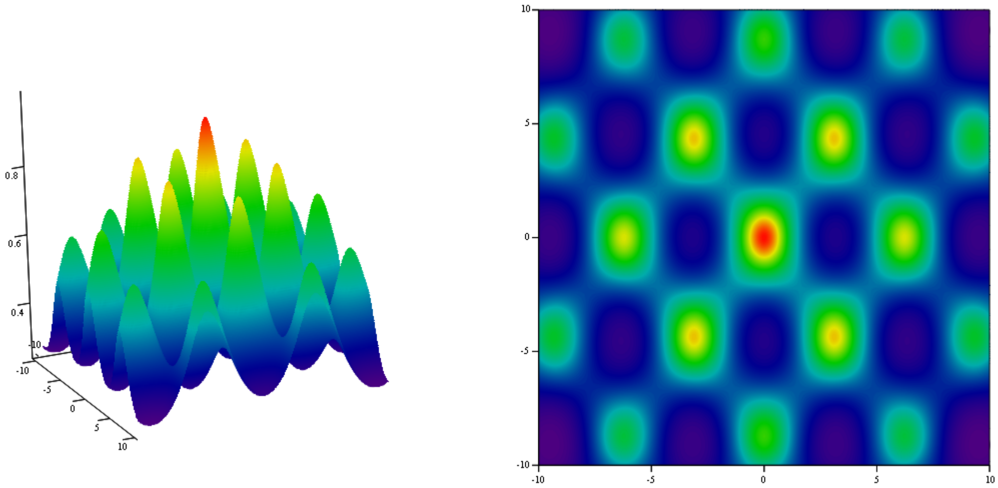

Рисунок 1.17. Функция ReverseGriewank

1.16.2 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:

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

1.16.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

1.16.4 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:

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

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

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:

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

1.16.5 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.

1.16.6 Реализация

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

Код 1.16. 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;
}
```

1.16.7 Ссылки

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

1.17 Функция «Лисьи норы» Шекеля

1.17.1 Описание функции

Идентификатор: MHL_TestFunction_ShekelsFoxholes.

Наименование: Функция «Лисьи норы» Шекеля.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = \frac{1}{\frac{1}{K} + \sum_{j=1}^{25} \frac{1}{j + (\bar{x}_1 - A_{1,j})^6 + (\bar{x}_2 - A_{2,j})^6}}, \text{ где} \quad (1.18)$$

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

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

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

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

График: Рисунок 1.18 нас 61 стр.

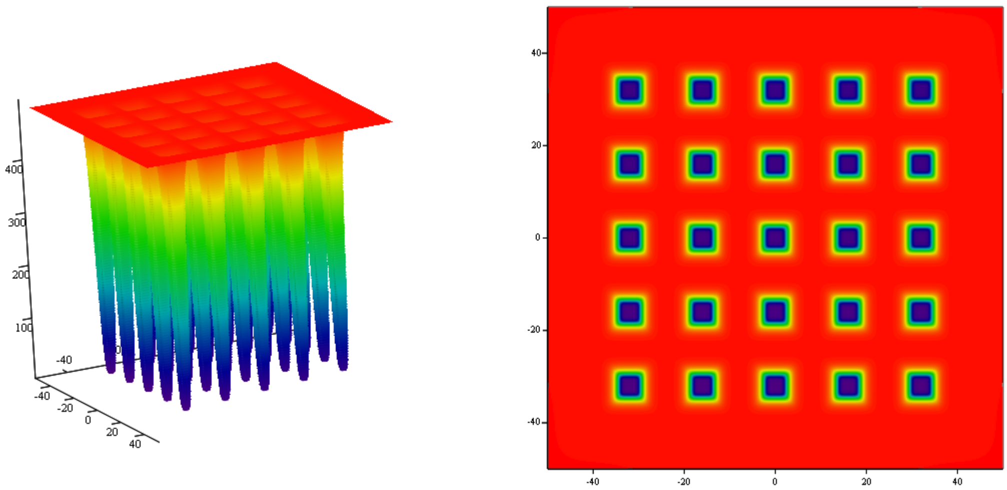

Рисунок 1.18. Функция «Лисьи норы» Шекеля

1.17.2 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:

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

1.17.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 2$.

1.17.4 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:

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

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

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:

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

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

1.17.6 Реализация

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

Код 1.17. 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;
}
```

1.17.7 Ссылки

Данная функция приводится в следующих источниках:

1. [12, стр. 34] — Эволюционные методы моделирования и оптимизации сложных систем.
2. [11, стр. 729] — International Conference on Intelligent Computing: Intelligent computing.

1.18 Функция Сомбреро

1.18.1 Описание функции

Идентификатор: MHL_TestFunction_Sombrero.

Наименование: Функция Сомбреро.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right)=\frac{1-sin\left(\sqrt{\bar{x}_1^2+\bar{x}_2^2}\right)^2}{1+0.001\left(\bar{x}_1^2+\bar{x}_2^2\right)},\text{ где} \tag{1.19}$$

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

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 1.19 нас 65 стр.

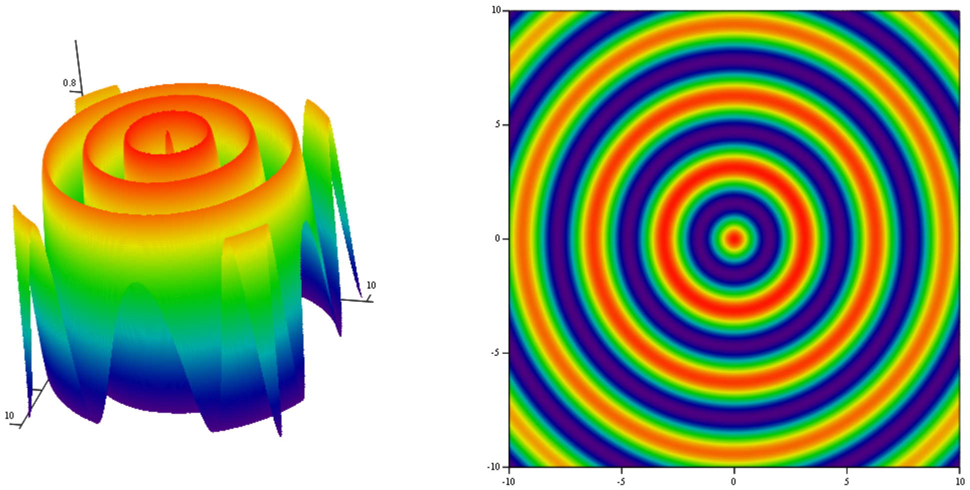

Рисунок 1.19. Функция Сомбреро

1.18.2 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:

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

1.18.3 Основная задача и подзадачи

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

1.18.4 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:

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

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

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:

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

1.18.5 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.

1.18.6 Реализация

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

Код 1.18. 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;
}
```

1.18.7 Ссылки

Данная функция приводится в следующих источниках:

1. [12, стр. 30] — Эволюционные методы моделирования и оптимизации сложных систем.

1.19 Функция Multiextremal

1.19.1 Описание функции

Идентификатор: MHL_TestFunction_Multiextremal.

Наименование: Multiextremal.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right) = 0.05\left(x-1\right)^2 + \left(3 - 2.9e^{-2.77257x^2}\right)\left(1 - \cos\left(x\left(4 - 50e^{-2.77257x^2}\right)\right)\right), \text{ где} \tag{1.20}$$

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

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

$n = 1$ — размерность вещественного вектора.

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

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

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

График: Рисунок 1.20 на с 68 стр.

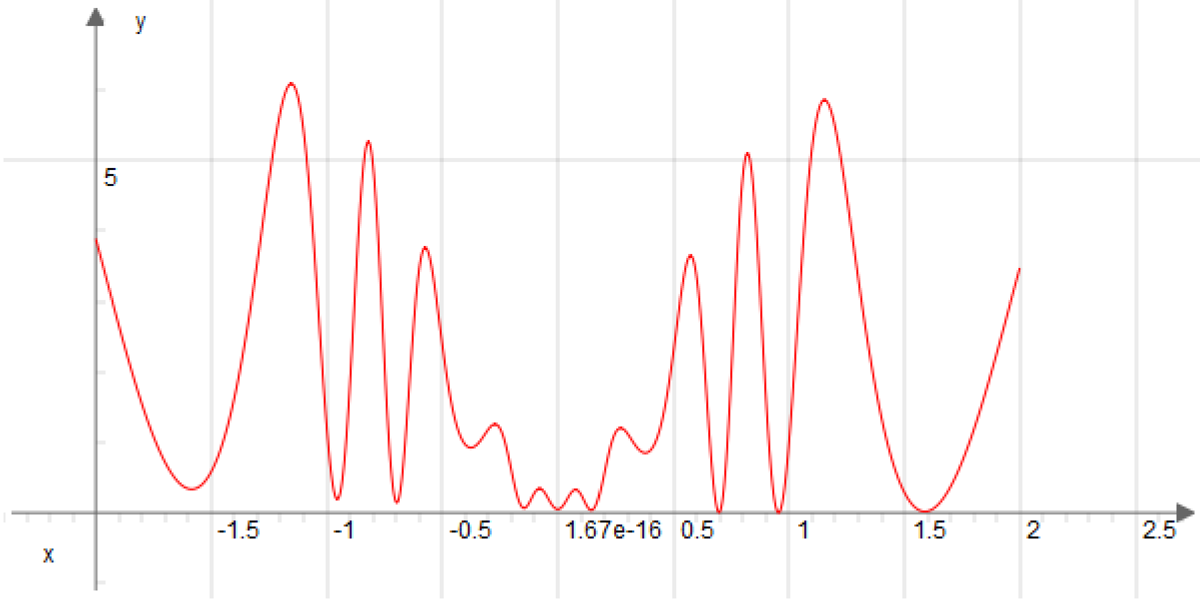

Рисунок 1.20. Функция Multiextremal

1.19.2 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:

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

1.19.3 Основная задача и подзадачи

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

1.19.4 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:

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

1.19.5 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.

1.19.6 Реализация

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

Код 1.19. 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;
}
```

1.19.7 Ссылки

Данная функция приводится в следующих источниках:

1. [12, стр. 26] — Эволюционные методы моделирования и оптимизации сложных систем.

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

1.20 Функция Multiextremal2

1.20.1 Описание функции

Идентификатор: MHL_TestFunction_Multiextremal2.

Наименование: Multiextremal2.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right) = 1 - 0.5\cos\left(1.5\left(10x - 0.3\right)\right)\cos\left(31.4x\right) + 0.5\cos\left(\sqrt{5}\cdot 10x\right)\cos\left(35x\right), \text{ где} \quad (1.21)$$

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

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

$n = 1$ — размерность вещественного вектора.

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

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

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

График: Рисунок 1.21 нас 71 стр.

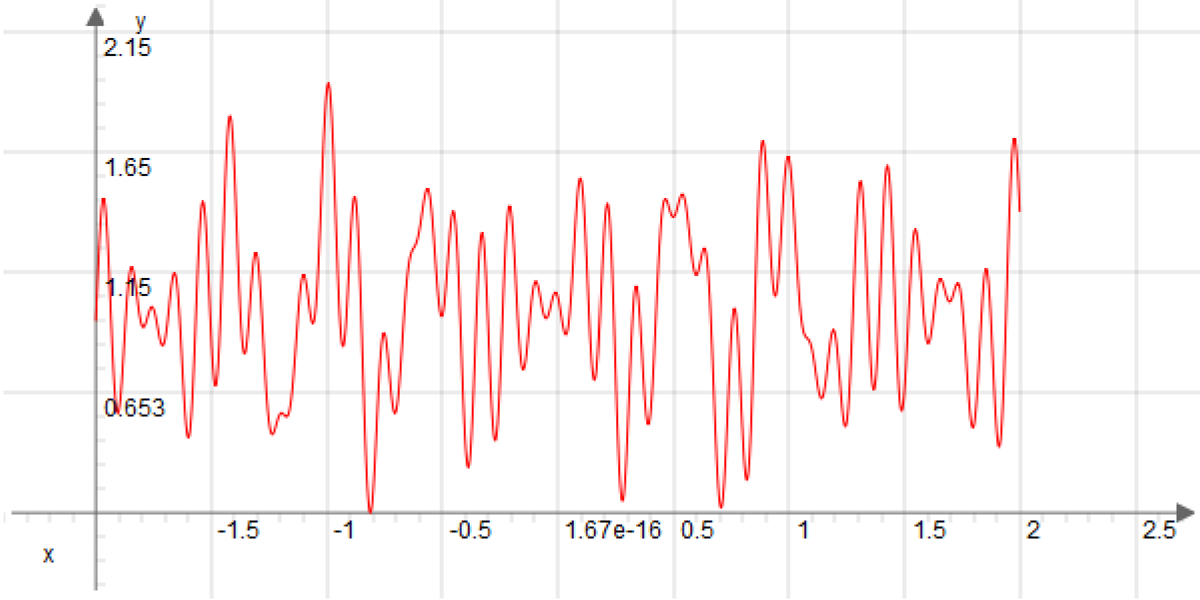

Рисунок 1.21. Функция Multiextremal2

1.20.2 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:

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

1.20.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 1$.

1.20.4 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:

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

1.20.5 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.

1.20.6 Реализация

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

Код 1.20. 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;
}
```

1.20.7 Ссылки

Данная функция приводится в следующих источниках:

1. [12, стр. 27] — Эволюционные методы моделирования и оптимизации сложных систем.

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

1.21 Функция волна

1.21.1 Описание функции

Идентификатор: MHL_TestFunction_Wave.

Наименование: Волна.

Тип: Задача вещественной оптимизации.

Формула (целевая функция):

$$f\left(\bar{x}\right)=e^{-\bar{x}_1^2}+0.01\cos\left(200\cdot\bar{x}_1\right),\text{ где} \quad (1.22)$$

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

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

$n=1$ — размерность вещественного вектора.

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

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

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

График: Рисунок 1.22 нас 74 стр.

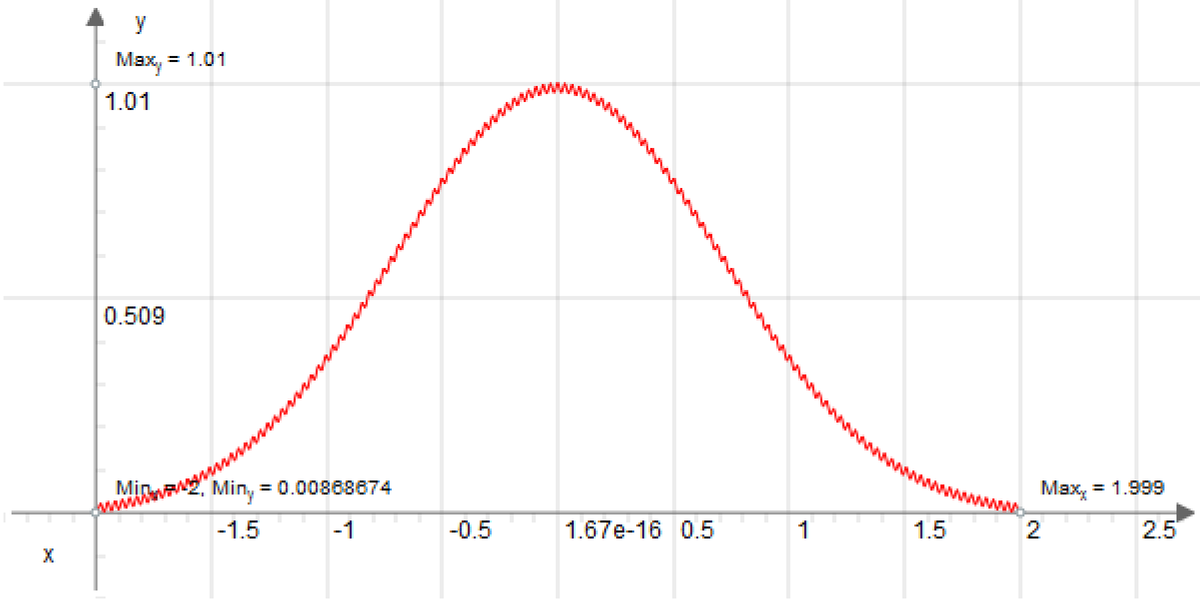

Рисунок 1.22. Волна

1.21.2 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:

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

1.21.3 Основная задача и подзадачи

Изменяемый параметр: n — размерность вещественного вектора.

Значение в основной задаче: $n = 1$.

1.21.4 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:

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

1.21.5 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.

1.21.6 Реализация

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

Код 1.21. 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;
}
```

1.21.7 Ссылки

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

Глава 2

Задачи бинарной оптимизации

2.1 Сумма всех элементов бинарного вектора

2.1.1 Описание функции

Идентификатор: MHL_TestFunction_SumVector.

Наименование: Сумма всех элементов бинарного вектора.

Тип: Задача бинарной оптимизации.

Формула (целевая функция):

$$f(\bar{x}) = \sum_{i=1}^{n} \bar{x}_i, \text{ где} \tag{2.1}$$

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

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

n — размерность бинарного вектора.

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

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

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

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

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

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

2.1.2 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:	$n = 200$.

2.1.3 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:

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

2.1.4 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.

2.1.5 Реализация

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

Код 2.1. 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. Paraboloid. http://en.wikipedia.org/wiki/Paraboloid.

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

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

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

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

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

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

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

11. 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.

12. 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.

13. 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.

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

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

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