Пантелеев Андрей Владимирович,
д.ф.-м.н., проф., заведующий кафедрой “Математическая кибернетика” МАИ

МЕТАЭВРИСТИЧЕСКИЕ АЛГОРИТМЫ ПОИСКА УСЛОВНОГО ЭКСТРЕМУМА

Постановка задачи

Дана целевая функция $f(x) = f(x_1, x_2, ..., x_n)$, определенная на множестве допустимых решений $D \subseteq R^n$.

Требуется найти глобальный условный максимум функции $f(x)$ на множестве D, т.е. такую точку $x^* \in D$, что

$$f(x^*) = \max_{x \in D} f(x), \tag{1.1}$$

где $x = (x_1, x_2, ..., x_n)^T$, $D = \{x \mid x_i \in [a_i, b_i], i = 1, 2, ..., n\}$.

Задача поиска минимума целевой функции $f(x)$ сводится к задаче поиска максимума путем замены знака перед функцией на противоположный:

$$f(x^*) = \min_{x \in D} f(x) = -\max_{x \in D}[-f(x)].$$

Метаэвристические алгоритмы:
- объединяют в себе один или более эвристических методов (процедур),
- опираются на стратегию поиска более высокого уровня (отсюда – ***мета***).

КЛАССИФИКАЦИЯ МЕТАЭВРИСТИЧЕСКИХ АЛГОРИТМОВ

I. ЭВОЛЮЦИОННЫЕ МЕТОДЫ (Evolutionary Methods) -

А. ***Генетические алгоритмы*** (Genetic Algorithms)

- ***генетические алгоритмы с бинарным кодированием***
- ***генетические алгоритмы с вещественным кодированием.***

Б. ***Методы, имитирующие иммунные системы организмов – методы искусственных иммунных систем*** (Artificial Immune Systems).

В. ***Метод рассеивания*** (Scatter Search).

Г. ***Эволюционная стратегия преобразования ковариационной матрицы*** (Covariance Matrix Adaptation Evolution Strategy)

Д. ***Метод динамических сеток*** (Variable Mesh Optimization).

Е. ***Метод дифференциальной эволюции*** (Differential Evolution).

Ж. ***Метод, имитирующий распространение сорняков*** (Weed Colonization) .

З. ***Метод, имитирующий поведение кукушек*** (Cuckoo Search).

II. МЕТОДЫ «РОЕВОГО» ИНТЕЛЛЕКТА -

А. ***Метод частиц в стае*** (Particle Swarm Optimization Strategy).

Б. ***Метод муравьиных колоний*** (Ant Colony Optimization) .

В. ***Метод имитации поведения бактерий*** (Bacterial Foraging Optimization).

Г. ***Методы пчелиных колоний*** (Bees Algorithms, Artificial Bee Colony):

- ***метод пчелиного роя***;
- ***метод искусственной пчелиной колонии***.

Д. ***Метод, имитирующий поведение стаи рыб*** в поисках корма (Fish School Search).

Е. ***Метод, имитирующий поведение летучих мышей*** (Bat-Inspired Algorithm).

Ж. ***Метод, имитирующий поведение светлячков*** (Glowwarm Swarm Optimization).

З. ***Алгоритм, имитирующий поведение лягушек*** (Shuffled Frog-Leaping Algorithm)

III. МЕТОДЫ, ИМИТИРУЮЩИЕ ФИЗИЧЕСКИЕ ПРОЦЕССЫ –

А. ***Метод гравитационной кинематики*** (Central Force Optimization).

Б. ***Метод имитации отжига*** (Simulated Annealing)

В. ***Адаптивном методе имитации отжига*** (Adaptive Simulated Annealing).

Г. ***Метод поиска гармонии*** (Harmony Search) .

Д. ***Метод, использующий закон электромагнетизма*** (Electromagnetism-like Mechanism).

IV. МУЛЬТИСТАРТОВЫЕ МЕТОДЫ –

А. Жадный адаптивный метод случайного поиска (Greedy Randomized Adaptive Search Procedure)

Б. Метод направленного табу-поиска (Tabu Search)

I. ЭВОЛЮЦИОННЫЕ МЕТОДЫ

А. ГЕНЕТИЧЕСКИЙ АЛГОРИТМ С БИНАРНЫМ КОДИРОВАНИЕМ

Стратегия поиска решения

ГА имитируют в своей работе природные способы оптимизации:
- генетическое наследование
- естественный отбор.

Целевая функция $f(x)$ эквивалентна природному понятию *приспособленности* живого организма.

Вектор параметров $x=(x_1,x_2,...,x_n)^T$ целевой функции называется *фенотипом*, а отдельные его параметры x_i – *признаками*, $i=1,2,...,n$.

Любой живой организм может быть представлен своим генотипом и фенотипом.

Генотип – это совокупность наследственных признаков, информация о которых заключена в хромосомном наборе.

Фенотип – совокупность всех признаков и свойств организма, формирующихся в процессе взаимодействия его генотипа и внешней среды. Каждый ген имеет свое отражение в фенотипе.

ГА ведут поиск решения только на уровне генотипа.

Каждую координату x_i вектора $x=(x_1,x_2,...,x_n)^T \in D$ представляют в некоторой форме s_i, удобной для использования в ГА и называемой *геном*. Для этого необходимо осуществить преобразование, в общем случае не взаимно однозначное, вектора параметров $x=(x_1,x_2,...,x_n)^T \in D$ в некоторую структуру $s=(s_1,s_2,...,s_n)^T \in S$, называемую *хромосомой (генотипом, особью)*:

$$D \xrightarrow{e} S,$$

где e – функция кодирования, S – пространство представлений (как правило, $D \neq S$).

Для того чтобы уметь восстановить по хромосоме решение, необходимо задать обратное преобразование:

$$S \xrightarrow{e^{-1}} D,$$

где e^{-1} – функция декодирования.

В пространстве представлений S вводится так называемая *функция приспособленности* $\mu(s)$: $S \xrightarrow{\mu} R$, где R – множество вещественных чисел, аналогичная целевой функции $f(x)$ на множестве D. Функцией $\mu(s)$ может быть любая функция, удовлетворяющая следующему условию:

$$\forall x^1,x^2 \in D: s^1=e(x^1),\ s^2=e(x^2),\ s^1 \neq s^2, \text{ если } f(x^1)>f(x^2), \text{ то } \mu(s^1)>\mu(s^2).$$

Решение исходной оптимизационной задачи $f(x^*) = \max_{x \in D} f(x)$ сводится к поиску решения s^* другой задачи оптимизации:

$$\mu(s^*) = \max_{s \in S} \mu(s) . \tag{1.2}$$

При решении используются конечные наборы

$$I = \{s^k = (s_1^k, s_2^k, ..., s_n^k)^T, \ k = 1, 2, ..., m\} \subset S$$

возможных решений, называемые *популяциями*, где s^k – хромосома с номером k, m – размер популяции, s_i^k – ген с номером i.

Затем осуществляется обратное преобразование:

$$x^* = e^{-1}(s^*) . \tag{1.3}$$

Бинарное кодирование.

- координате x_i ставится в соответствие целое число $\beta_i \in \{0, 1, ..., k_i\}$:

$$\beta_i = \begin{cases} 0, & x_i = a_i, \\ \left[\dfrac{x_i - a_i}{h_i}\right] + 1, & x_i \in (a_i, b_i), \\ k_i, & x_i = b_i, \end{cases}$$

где $[\cdot]$ означает операцию выделения целой части числа.

- каждому номеру $\{0, 1, ..., k_i\}$ ставится в соответствие уникальная бинарная комбинация длины l_i. Величина l_i определяется из условия: $k_i = 2^{l_i} - 1$. Преобразование в битовую строку осуществляется с помощью обычного двоичного кодирования (позиционного кода) или *рефлексивного кода Грея*, который обладает свойством непрерывности бинарной комбинации: изменению кодируемого числа β_i на единицу соответствует изменение бинарной комбинации только в одном разряде.

Декодирование. Значение координаты x_i при $\gamma_i = 0$ определяется левым концом промежутка $[a_i, b_i]$: $x_i = a_i$, при $\gamma_i = k_i$ – правым концом: $x_i = b_i$, а при $\gamma_i = 1, ..., k_i - 1$ – серединой соответствующего подынтервала с номером γ_i :

$$x_i = \begin{cases} a_i, & \gamma_i = 0, \\ \dfrac{(a_i + (\gamma_i - 1)h_i) + (a_i + \gamma_i h_i)}{2}, & \gamma_i = 1, 2, ..., k_i - 1, \\ b_i, & \gamma_i = k_i. \end{cases}$$

Общая схема работы генетического алгоритма (ГА)

Алгоритм решения задачи

Шаг 1. *Создание начальной популяции.*

Шаг 1.1. Задается номер популяции $t=0$, максимальное количество популяций $t_{\max}$, номер итерации $M=1$, параметр кодирования l_i, $i=1,...,n$ (т.е. длина бинарной комбинации).

Шаг 1.2. Задается размер популяции m. Случайным образом формируется начальная популяция I_0. Для этого с помощью равномерного распределения на единичном отрезке $[0,1]$ m раз генерируем последовательность из n случайных точек $\{P_i^{0,k}\}_{i=1}^{n}, k=1,2,...,m$. Используя линейное преобразование, отображаем каждую точку на соответствующий ей промежуток $[a_i,b_i]$: $P_i^k=(b_i-a_i)P_i^{0,k}+a_i$, $i=1,2,...,n$; $k=1,2,...,m$. Составляя векторы из точек последовательности $\{P_i^k\}_{i=1}^{n}$ при фиксированных k, получаем m начальных векторов $x^k=(x_1{}^k,x_2{}^k,...,x_n{}^k)^T$, $x_i{}^k=P_i^k$, $i=1,2,...,n$.

Далее, разбиваем промежутки $[a_i,b_i]$ на подынтервалы в соответствии с параметром кодирования l_i. Определяем, каким из них принадлежат значения $x_i{}^k$, то есть, находим $\beta_i{}^k$. Числа $\beta_i{}^k$ кодируем с помощью кода Грея, преобразуя их в битовые строки $s_i{}^k$, из которых составляем m хромосом вида $s^k=(s_1{}^k,s_2{}^k,...,s_n{}^k)^T$, $s_i{}^k \leftrightarrow \beta_i^k$, $i=1,2,...,n$.

В результате описанного процесса сформирована начальная популяция $I_0=\{s^k=(s_1^k,s_2^k,...,s_n^k)^T, k=1,2,...,m\}\subset S$.

Шаг 1.3. Вычисляется значение функции приспособленности для каждой особи $s^k\in I_0$: $\mu_k=\mu(s^k)$, $k=1,2,...,m$, и популяции I_0 в целом: $\mu=\sum_{k=1}^{m}\mu_k$.

Шаг 2. *Селекция.*

Селекция – это операция, которая осуществляет отбор особей (хромосом) s^k в соответствии со значениями функции приспособленности $\mu(s^k)$ для последующего их скрещивания. Вероятность участия в скрещивании обычно пропорциональна относительной приспособленности $\frac{\mu(s^k)}{\mu}$ особи s^k.

Виды селекции.

1) *Панмиксия* (*случайный равновероятный отбор*). Особи – «родители» случайным образом выбираются из всей популяции, причем любая может стать членом нескольких пар. Несмотря на простоту, такой подход универсален для решения различных классов задач. Вероятность участия особи в селекции составляет $p_k = \frac{1}{m}$.

2) *Селективный* (*рангово-пропорциональный*) *отбор*. Его суть состоит в том, что «родителями» могут стать только те особи, значения функций приспособленности которых не меньше среднего значения функции приспособленности по популяции, то есть при выполнении условия $\mu(s^k) \geq \frac{\mu}{m}$, при равной вероятности таких кандидатов составить брачную пару.

3) *Рулетка*. Отбирает особей с помощью σ «запусков» рулетки (если выбирается пара родителей, то $\sigma = 2$). Колесо рулетки содержит по одному сектору для каждого члена популяции. Размер k-го сектора пропорционален соответствующей величине p_k, вычисляемой по формуле $p_k = \frac{\mu(s^k)}{\mu}$.

4) *Турнирный отбор*. Реализует σ турниров, чтобы выбрать σ особей (если выбирается пара родителей, то $\sigma = 2$). В каждом турнире случайным образом выбирается v элементов из популяции, а затем лучшая особь среди них. Наиболее распространен турнирный отбор с $v = 2$.

5) *Инбридинг*. Первый член родительской пары выбирается случайно, а вторым с большей вероятностью будет максимально «близко родственная» к нему особь. Под родством здесь понимается *расстояние Хэмминга* между членами популяции. Расстояние Хэмминга d_h – это число несовпадающих по своим значениям битов в битовых строках: $d_h(s^k, s^p) = \sum_{i=1}^{n} | s_i^k - s_i^p |$. Особи s^k, s^p будут считаться близкими, если $d_h(s^k, s^p) \leq d_0$, где $d_0 \in N$ – параметр метода, задаваемый пользователем.

6) *Аутбридинг*. В противоположность инбридингу формирует брачные пары из максимально «далеких» особей.

В ГА может применяться так называемая *стратегия элитизма*. Ее суть в том, что небольшое количество особей s^k (как правило, одна или две) переходит в следующее поколение без изменений, не участвуя в селекции и последующем скрещивании.

Результатом шага 2 являются две особи s^1 и s^2, выбранные в качестве родительской пары с помощью одного из операторов селекции.

Шаг 3. *Скрещивание.*

Скрещивание – это операция, при которой из нескольких, обычно двух хромосом (особей), называемых родителями, порождается одна или несколько новых, называемых потомками, путем обмена частями родительских хромосом. В простейшем случае скрещивание в ГА реализуется так же, как и в биологии: хромосомы разрезаются в случайной точке (точках) и обмениваются частями между собой.

Виды скрещивания.

1) *Одноточечное скрещивание*. Сначала случайным образом формируется одна точка разрыва. Точка разрыва – участок между соседними битами в строке (может разрывать ген). Обе родительские структуры разрываются на два сегмента по этой точке. Затем соответствующие сегменты различных родителей «склеиваются» и получаются два потомка.

2) *Многоточечное скрещивание*. Выбираются две (или более) точки разрыва, и родительские хромосомы обмениваются сегментом, находящимся между этими точками.

3) *Равномерное скрещивание*. Первый потомок наследует каждый бит первого родителя с заданной вероятностью p_j, а каждый бит второго родителя с вероятностью $1-p_j$. При этом из двух битов, находящихся в одинаковых позициях в родительских хромосомах, бит, не унаследованный первым потомком, наследуется вторым.

Результатом шага 3 являются два потомка s^{ch1} и s^{ch2}, полученные путем применения одного из операторов скрещивания к хромосомам - родителям s^1 и s^2.

Шаг 4. *Мутация.*

Мутация – это преобразование хромосомы, случайно изменяющее один или несколько из ее генов. Оператор мутации предназначен для того, чтобы поддерживать разнообразие особей в популяции. При использовании данного оператора каждый бит в хромосоме с определенной вероятностью $p_j \in (0,1)$, $j = 1,...,l$, инвертируется, то есть нуль меняется на единицу (единица меняется на нуль).

Вместо оператора мутации может быть использован *оператор инверсии*, действие которого заключается в том, что хромосома делится на две части в случайной точке разрыва, а полученные части меняются затем местами.

Результатом шага 4 являются два потомка-мутанта s^{M1} и s^{M2}, полученные путем применения оператора мутации или инверсии к хромосомам-потомкам s^{ch1} и s^{ch2}.

Шаг 5. *Формирование новой популяции.*

Пусть в результате последовательного применения к хромосомам-родителям s^1 и s^2 операторов скрещивания и мутации на предыдущем шаге были получены потомки-мутанты s^{M1} и s^{M2}.

С вероятностью 0,5 выбирается один из потомков s^{M1} и s^{M2}. Пусть это будет s^{M1}. Выбранный потомок добавляется в популяцию вместо хромосомы, которой соответствует наименьшее среди элементов популяции значение функции приспособленности, вычисляется значение функции приспособленности $\mu_z = \mu(s^{M1})$.

Проверяются условия:

а) если $M < m$, то положить $M = M + 1$ и перейти к шагу 2;

б) если $M = m$, то положить $t = t + 1$ и перейти к шагу 6.

Шаг 6. *Проверка условий окончания работы ГА.*

Условием окончания работы ГА является формирование заданного количества популяций: $t = t_{\max}$;

а) если условие не выполнено, то полагаем $M = 1$ и переходим к шагу 2;

б) если условие окончания работы выполнено, то в качестве решения (приближенного) задачи (1.2) выбирается особь с лучшим значением функции приспособленности из текущей популяции: $s^* \cong \tilde{s}^* = \arg\max\limits_{k=1,\dots,m} \mu(s^k)$, а по нему с помощью операции декодирования (1.3) определяется приближенное решение поставленной задачи (1.1): $x^* \cong \tilde{x}^* = e^{-1}(\tilde{s}^*)$.

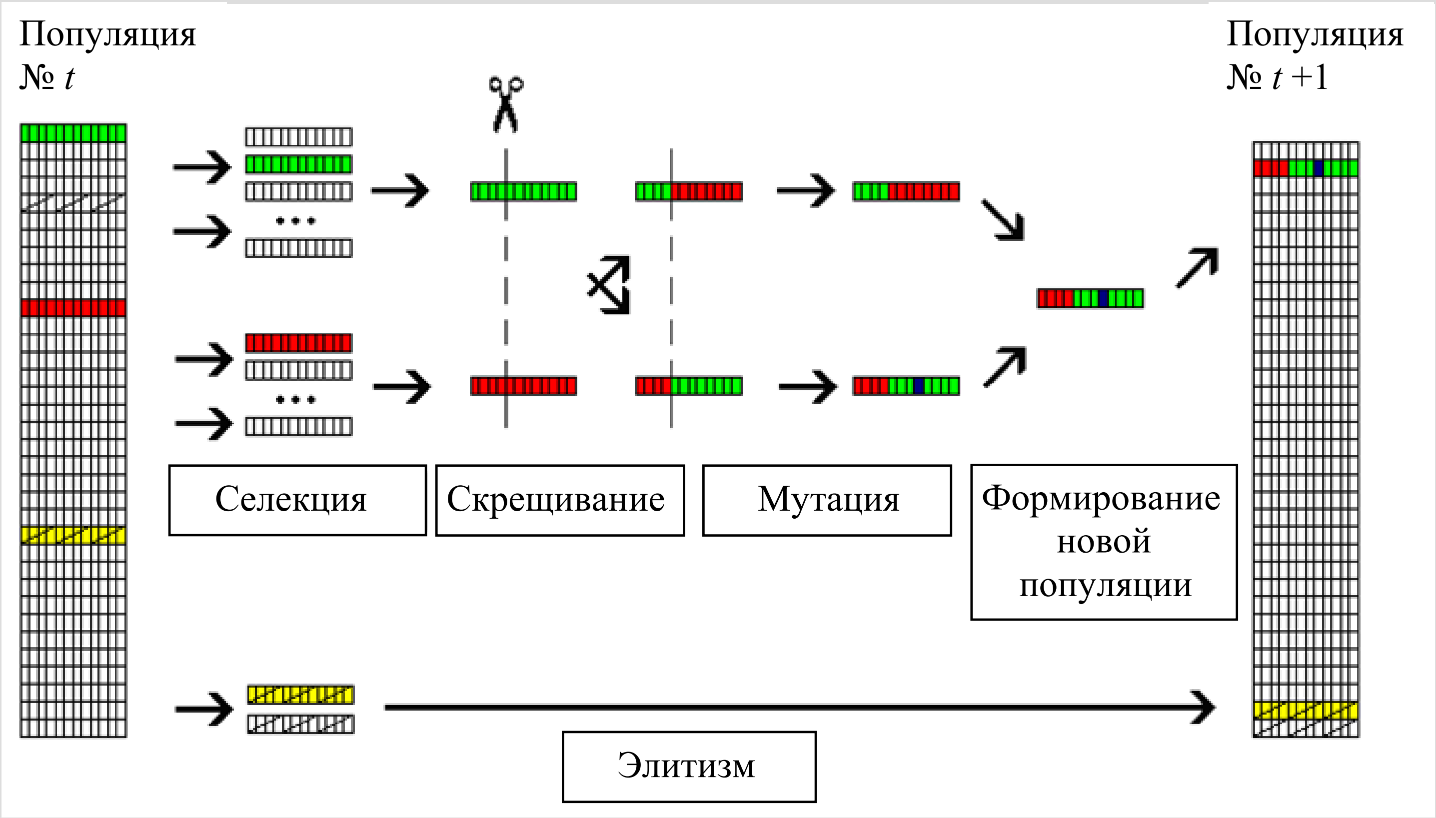


Схема формирования новой популяции

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ С ВЕЩЕСТВЕННЫМ КОДИРОВАНИЕМ

Стратегия поиска решения

Рассматриваемая в задаче (1.1) *целевая функция* $f(x)$ эквивалентна природному понятию *приспособленности* живого организма.

Вектор параметров $x=(x_1,x_2,...,x_n)^T$ целевой функции называется *фенотипом*, а отдельные его координаты $x_i \in R$ – *признаками*, $i=1,2,\dots,n$. Любой вектор $x=(x_1,x_2,...,x_n)^T \in D$ считается *хромосомой* (*генотипом, особью*), а каждая его координата $x_i \in R$ представляет собой *ген*. Фенотип совпадает с генотипом, а поиск фактически производится в пространстве фенотипов.

На множестве допустимых решений D вводится так называемая *функция приспособленности* $\mu(x)$: $D \xrightarrow{\mu} R$, где R – множество вещественных чисел, аналогичная целевой функции $f(x)$.

Функцией $\mu(x)$ может быть любая функция, удовлетворяющая следующему *условию*:

$$\forall x^1, x^2 \in D: \ x^1 \neq x^2, \text{ если } f(x^1) > f(x^2), \text{ то } \mu(x^1) > \mu(x^2).$$

Решение исходной оптимизационной задачи $f(x^*) = \max\limits_{x \in D} f(x)$ сводится к поиску решения x_μ^* другой оптимизационной задачи:

$$\mu(x_\mu^*) = \max_{x \in D} \mu(x). \qquad (1.4)$$

В силу выбора функции $\mu(x)$, решения задач (1.1) и (1.4) (хромосома) совпадают:

$$x^* = \arg\max_{x \in D} f(x) = \arg\max_{x \in D} \mu(x) = x_\mu^*. \qquad (1.5)$$

При решении задачи (1.4) используются конечные наборы $I = \{x^k = (x_1^k, x_2^k, \dots, x_n^k)^T, \ k=1,2,\dots,m\} \subset D$ возможных решений, называемые *популяциями*, где x^k – хромосома с номером k, m – размер популяции, x_i^k – ген с номером i.

Алгоритм решения задачи

Шаг 1. *Создание начальной популяции.*

Шаг 1.1. Задается номер популяции $t=0$, максимальное количество популяций $t_{\max}$, номер итерации цикла $M=1$; размер популяции m.

Шаг 1.2. Случайным образом формируется начальная популяция I_0. Для этого с помощью равномерного распределения на единичном отрезке $[0,1]$ m раз генерируем последовательность из n случайных точек $\{P_i^{0,k}\}_{i=1}^{n}$, $i=1,...,n,$ $k=1,2,\ldots,m$. Используя линейное преобразование, каждая точка отображается на соответствующий ей промежуток $[a_i,b_i]$: $P_i^k=(b_i-a_i)P_i^{0,k}+a_i$. Составляя векторы из точек последовательности $\{P_i^k\}_{i=1}^{n}$ при фиксированных k, получаем m начальных векторов $x^k=(x_1{}^k,x_2{}^k,\ldots,x_n{}^k)^T$, $x_i{}^k=P_i^k$, $i=1,2,\ldots,n$, координаты которых x_i имеют равномерное распределение на отрезках $[a_i,b_i]$, $i=1,...,n$. Таким образом, может быть сформирована начальная популяция $I_0=\{x^k, k=1,2,\ldots,m \mid x^k=(x_1^k,x_2^k,\ldots,x_n^k)^T\in D\}$.

Шаг 1.3. Вычисляется значение функции приспособленности для каждой особи $x^k\in I_0$: $\mu_k=\mu(x^k)$, $k=1,2,...,m$, и популяции I_0 в целом: $\mu=\sum_{k=1}^{m}\mu_k$.

Шаг 2. *Селекция.*

Селекция – это операция, которая осуществляет отбор особей (хромосом) x^k в соответствии со значениями функции приспособленности $\mu(x^k)$ для последующего их скрещивания. Вероятность участия в скрещивании обычно пропорциональна относительной приспособленности $\frac{\mu(x^k)}{\mu}$ особи x^k. Операторы селекции ГА с бинарным кодированием используются и в ГА с вещественным кодированием.

Виды селекции.

1) *Панмиксия* (*случайный равновероятный отбор*). Особи-«родители» случайным образом выбираются из всей популяции, причем любая может стать членом нескольких пар. Вероятность участия особи в селекции составляет $p_k = \frac{1}{m}$.

2) *Рулетка.* Отбирает особей с помощью σ «запусков» рулетки (если выбирается пара родителей, то $\sigma = 2$). Колесо рулетки содержит по одному сектору для каждого члена популяции. Размер k-го сектора пропорционален соответствующей величине p_k, вычисляемой по формуле: $p_k = \frac{\mu(x^k)}{\mu}$.

3) *Турнирный отбор.* Реализует σ турниров, чтобы выбрать σ особей (если выбирается пара родителей, то $\sigma = 2$). В каждом турнире случайным образом выбирается v элементов из популяции, а затем лучшая особь среди них. Наиболее распространен турнирный отбор с $v = 2$.

В ГА может применяться так называемая *стратегия элитизма*. Ее суть в том, что небольшое количество особей x^k (как правило, одна или две) переходит в следующее поколение без изменений, не участвуя в селекции и последующем скрещивании.

Результатом шага 2 являются две особи x^1 и x^2, выбранные в качестве родительской пары с помощью одного из операторов селекции.

Шаг 3. *Скрещивание.*

Скрещивание – это операция, при которой из нескольких, обычно двух хромосом (особей), называемых родителями, порождается одна или несколько новых, называемых потомками.

Пусть имеются две родительские хромосомы $x^1 = (x_1^1, x_2^1, \ldots, x_n^1)^T$, $x^2 = (x_1^2, x_2^2, \ldots, x_n^2)^T$, выбранные одним из операторов селекции.

Виды скрещивания.

1) *Плоский кроссовер.* Создается один потомок $x^q = (x_1^q, ..., x_n^q)^T$, где $x_i^q, i = 1, ..., n$ – случайное число из промежутка $[x_i^1, x_i^2]$. Здесь считается, что $x_i^2 \geq x_i^1$. Если это неравенство не выполняется, значения меняются местами.

2) *Простейший кроссовер.* Из множества $\{1, 2, ..., n-1\}$ выбирается случайное число p и генерируются два потомка:

$$x^{q1} = (x_1^1, ..., x_p^1, x_{p+1}^2 \ldots, x_n^2)^T;$$
$$x^{q2} = (x_1^2, ..., x_p^2, x_{p+1}^1 \ldots, x_n^1)^T.$$

3) *Арифметический кроссовер*. Создаются два потомка $x^{q1} = (x_1^{q1}, ..., x_n^{q1})^T$, $x^{q2} = (x_1^{q2}, ..., x_n^{q2})^T$:

$$x_i^{q1} = \eta\, x_i^1 + (1 - \eta)\, x_i^2,$$
$$x_i^{q2} = \eta\, x_i^2 + (1 - \eta)\, x_i^1,$$

где η – константа из интервала $(0;1)$, $i = 1, 2, ..., n$.

4) *Геометрический кроссовер*. Создаются два потомка x^{q1}, x^{q2}, где

$$x_i^{q1} = (\, x_i^1\,)^{\eta} \cdot (\, x_i^2\,)^{1-\eta},$$
$$x_i^{q2} = (\, x_i^2\,)^{\eta} \cdot (\, x_i^1\,)^{1-\eta},$$

где η – случайное число из интервала $(0;1)$, $i = 1, 2, ..., n$. Заметим, что при отрицательных значениях генов родителей могут возникать вычислительные проблемы.

5) *Дискретный кроссовер*. Создается один потомок $x^q = (x_1^q, ..., x_n^q)^T$, где x_i^q выбирается из двух значений x_i^1, x_i^2 случайно с вероятностью $0,5$ для всех $i = 1,...,n$.

6) *Смешанный кроссовер*. Генерируется один потомок x^q, где $x_i^q, i = 1,...,n$ – случайное число из промежутка $[x_{i\,\min} - \Delta\,\alpha, x_{i\,\max} + \Delta\,\alpha]$, $x_{i\,\min} = \min\{x_i^1, x_i^2\}$, $x_{i\,\max} = \max\{x_i^1, x_i^2\}$, $\Delta = x_{i\,\max} - x_{i\,\min}$, где α – параметр (рекомендуется $\alpha = 0,5$). При $\alpha = 0$ смешанный кроссовер превращается в плоский.

7) *Линейный кроссовер*. Создаются три потомка x^{q1}, x^{q2}, x^{q3}, где $x_i^{q1} = \dfrac{x_i^1 + x_i^2}{2}$, $x_i^{q2} = \dfrac{3x_i^1 - x_i^2}{2}$, $x_i^{q3} = \dfrac{-x_i^1 + 3x_i^2}{2}$, $i = 1,...,n.$

8) *Расширенный линейный кроссовер*. Создается один потомок x^q, где $x_i^q = x_i^1 + \eta \cdot (x_i^2 - x_i^1)$, $i = 1,...,n$, где η – случайное число из промежутка $[-0,25; 1,25]$.

9) *Кроссовер, имитирующий двоичный*. Создаются два потомка x^{q1}, x^{q2}, где

$$x_i^{q1} = \frac{1}{2}[(1-\eta^1)\,x_i^1 + (1+\eta^1)\,x_i^2\,],$$

$$x_i^{q2} = \frac{1}{2}[(1+\eta^2)\,x_i^1 + (1-\eta^2)\,x_i^2\,],$$

где η^j, $j = 1,2$ – числа, полученные по формуле

$$\eta^j = \begin{cases} (2u)^{\frac{1}{C+1}}, & u \le 0,5, \\ \left(\dfrac{1}{2(1-u)}\right)^{\frac{1}{C+1}}, & u > 0,5, \end{cases} \qquad j = 1,2\,,$$

u – случайное число, равномерно распределенное на $(0;1)$, $C \in [2;5]$ – параметр.

Результатом шага 3 являются один, два или три потомка x^{q1}, x^{q2}, x^{q3}, полученные после применения одного из операторов скрещивания к хромосомам - родителям x^1 и x^2.

Координаты хромосом-потомков должны принадлежать множеству допустимых решений. Если полученная в результате скрещивания (при помощи смешанного, линейного, линейного расширенного кроссоверов) i-я координата не принадлежит отрезку $[a_i, b_i]$, то она заменяется случайным числом, равномерно распределенным на $[a_i, b_i]$.

Шаг 4. *Мутация.*

Мутация – это преобразование хромосомы, случайно изменяющее один или несколько из ее генов. Оператор мутации предназначен для того, чтобы поддерживать разнообразие особей в популяции.

Виды мутации.

1) *Случайная.* Поочередно рассматривается каждый потомок, полученный в результате скрещивания. Среди генов $x_1,...,x_n$ случайно (с вероятностью $\frac{1}{n}$) выбирается один с номером $p \in \{1,...,n\}$, подлежащий замене. Его новое значение x_p^M случайным образом выбирается из промежутка $[a_p, b_p]$ изменения выбранной координаты x_p.

2) *Неравномерная.* Поочередно рассматривается каждый потомок, полученный в результате скрещивания. Среди генов $x_1,...,x_n$ случайно (с вероятностью $\frac{1}{n}$) выбирается один, подлежащий замене:

$$x_p^M = \begin{cases} x_i + \delta(t, (b_i - x_i)), & u = 0, \\ x_i - \delta(t, (x_i - a_i)), & u = 1, \end{cases}$$

где u – случайное число, выбранное с вероятностью 0,5 из множества $\{0; 1\}$; $\delta(t, y)$ – функция, определяющая значение на промежутке $[0, y]$:

$$\delta(t, y) = y\left(1 - r^{\left(1 - \frac{t}{t_{\max}}\right)C}\right),$$

r – случайное на отрезке $[0, 1]$ число, $t_{\max}$ – максимальное число итераций, C – параметр оператора.

Результатом шага 4 являются потомки, найденные на шаге 3, с одним измененным геном каждый: x^{M1}, x^{M2}, x^{M3}.

Шаг 5. *Формирование новой популяции.*

С равной вероятностью из потомков - мутантов x^{M1}, x^{M2}, x^{M3} предыдущего шага выбирается один. Пусть это будет x^{M1}**.** Выбранный потомок добавляется в популяцию вместо хромосомы, которой соответствует наименьшее среди элементов популяции значение функции приспособленности, вычисляется значение функции приспособленности $\mu_{M1} = \mu(x^{M1})$.

Проверяются условия:

а) если $M < m$, то положить $M = M + 1$ и перейти к шагу 2;

б) если $M = m$, то положить $t = t + 1$ и перейти к шагу 6.

Шаг 6. *Проверка условий окончания работы ГА.*

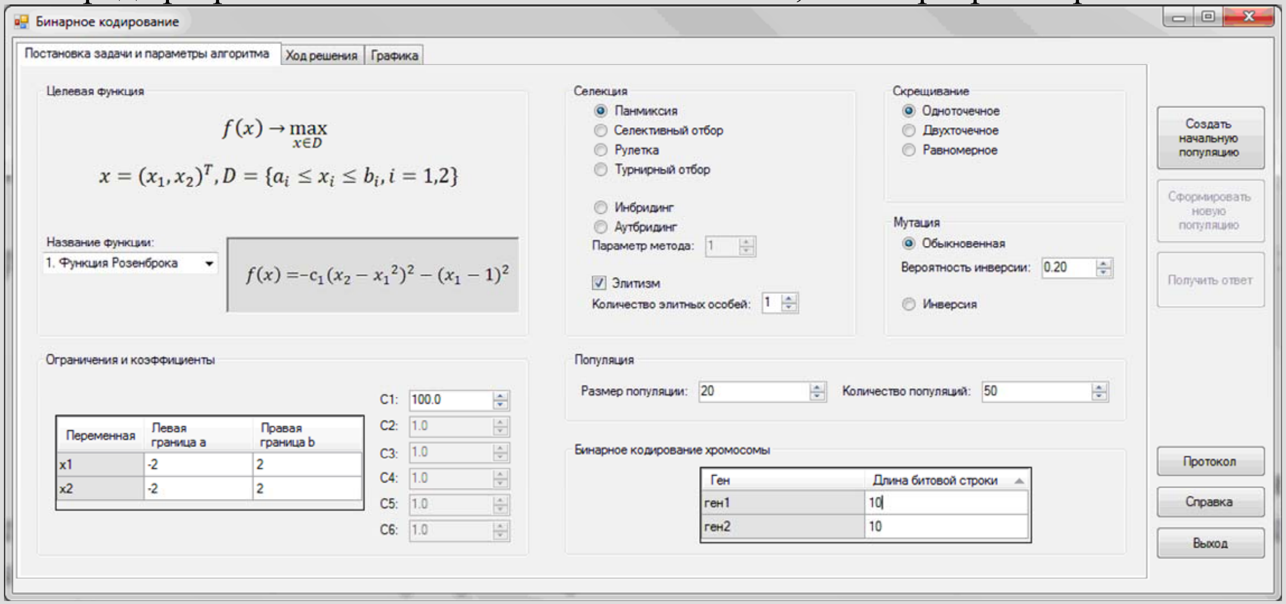
Условием окончания работы ГА является формирование заданного количества популяций: $t = t_{\max}$;

а) если условие не выполнено, то полагаем $M = 1$ и переходим к шагу 2;

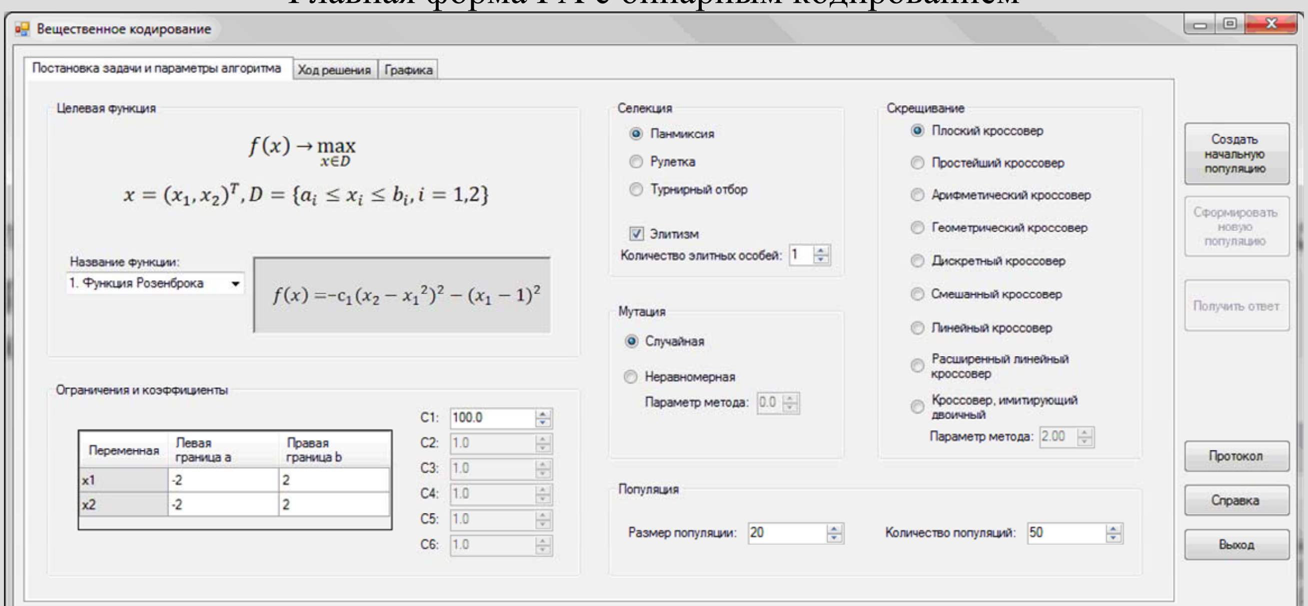
б) если условие окончания работы выполнено, то в качестве решения (приближенного) задачи $\mu(x_\mu^*) = \max\limits_{x \in D} \mu(x)$ выбирается особь с лучшим значением функции приспособленности из текущей популяции: $x_\mu^* \cong \tilde{x}_\mu^* = \arg\max\limits_{k=1,\dots,m} \mu(x^k)$, а по нему определяется приближенное решение поставленной задачи $f(x^*) = \max\limits_{x \in D} f(x)$: $x^* = x_\mu^*$.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

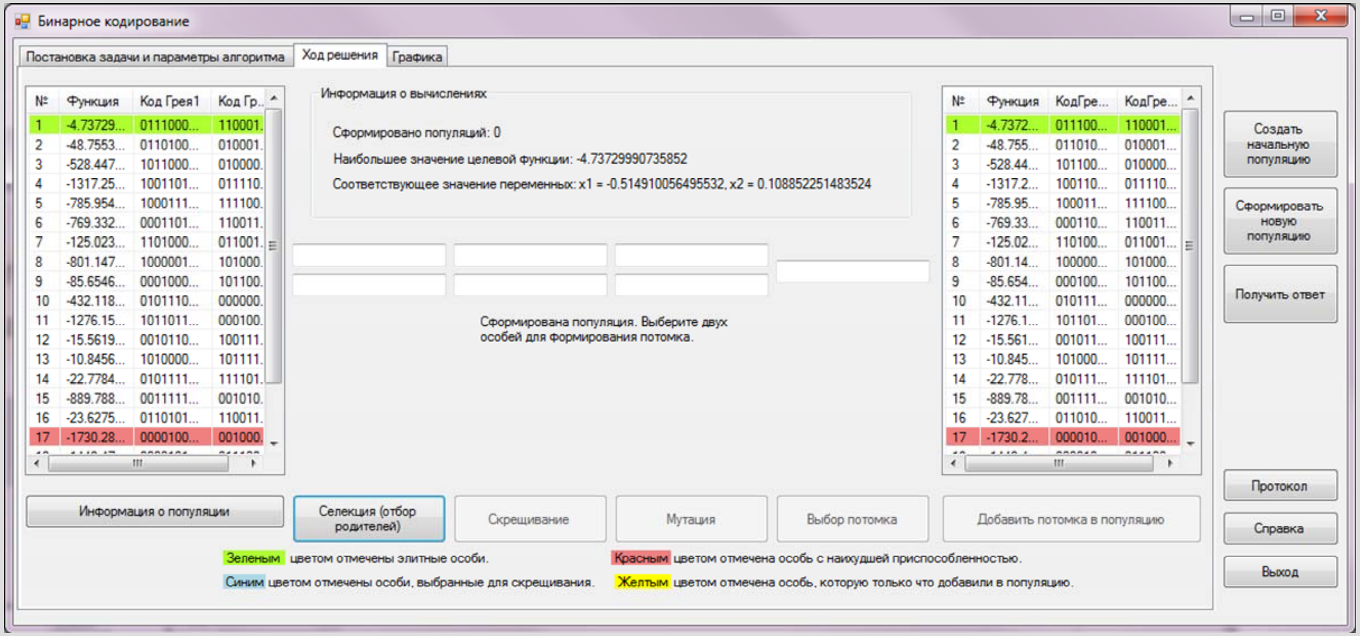
Среда разработки Microsoft Visual Studio 2005, язык программирования C# .



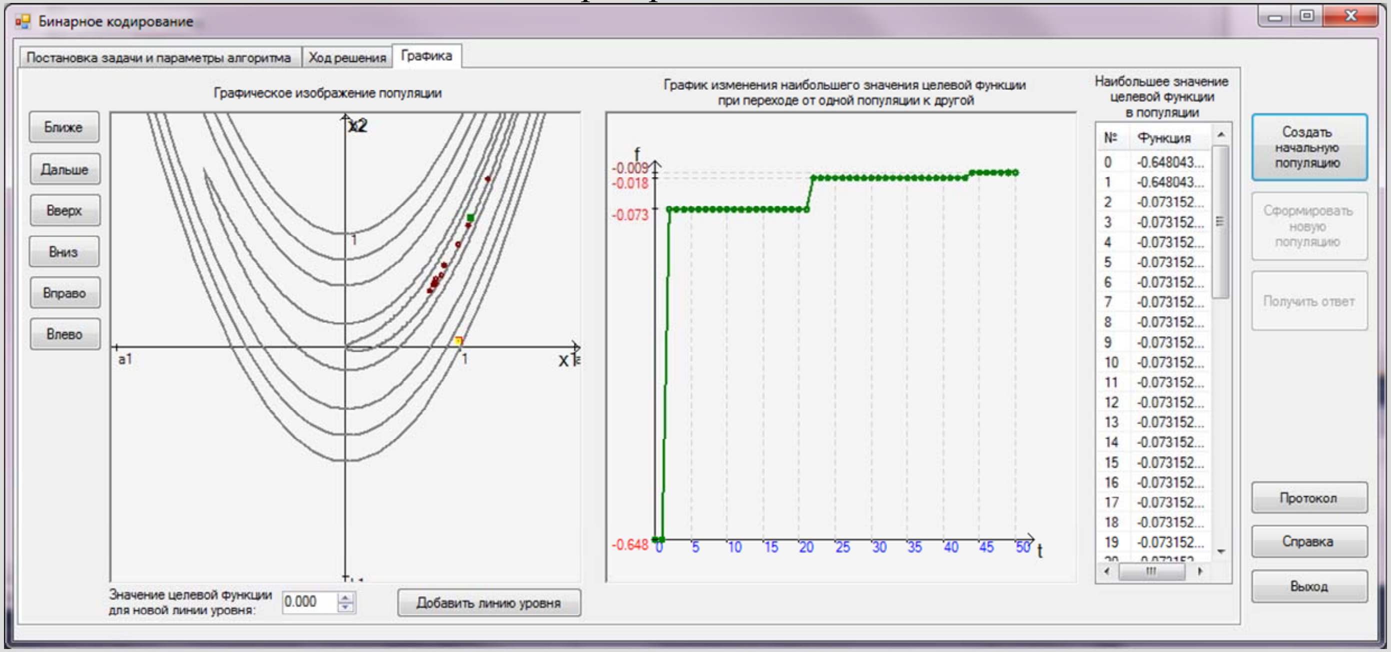
Главная форма ГА с бинарным кодированием



Главная форма ГА с вещественным кодированием



Форма работы по шагам

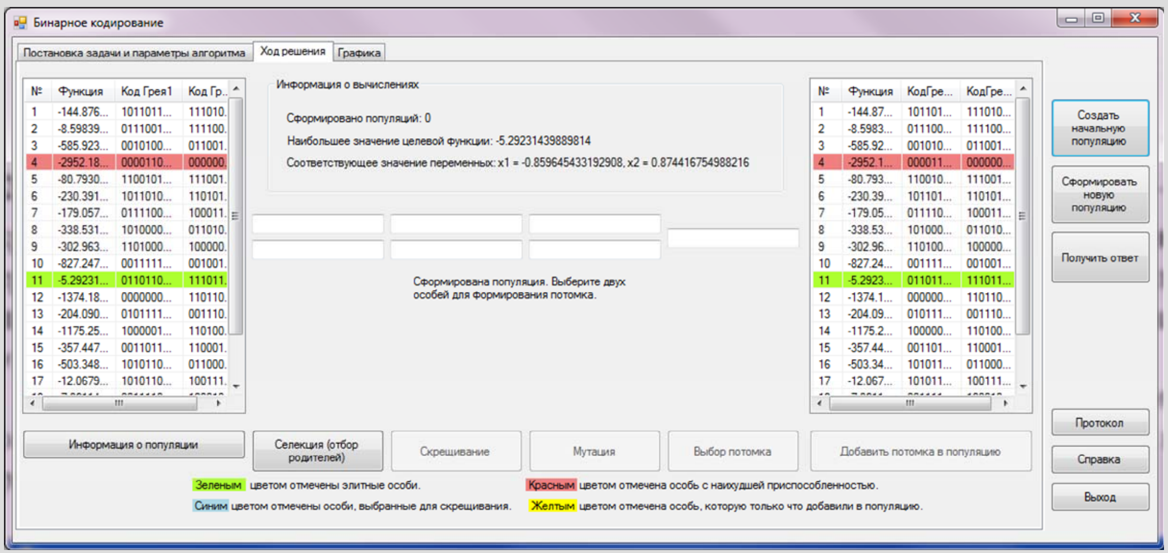


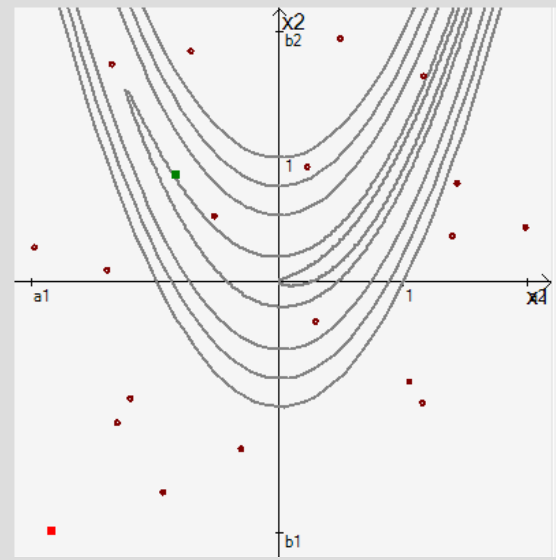
Графическая информация

Пример. Продемонстрируем работу ***генетического алгоритма с бинарным кодированием*** на примере поиска условного максимума функции Розенброка (табл. П.1).

Зададим параметры алгоритма: область допустимых решений $-2 \le x_1 \le 2$, $-2 \le x_2 \le 2$, метод селекции – панмиксия, количество элитных особей – 1, метод скрещивания – одноточечное скрещивание, метод мутации – обыкновенная с вероятностью инверсии $p = 0,2$, размер популяции – $m = 20$ особей, количество популяций – $t_{\max} = 50$, длина битовой строки первого гена – $l_1 = 10$, длина битовой строки второго гена – $l_2 = 10$,– и создадим начальную популяцию.

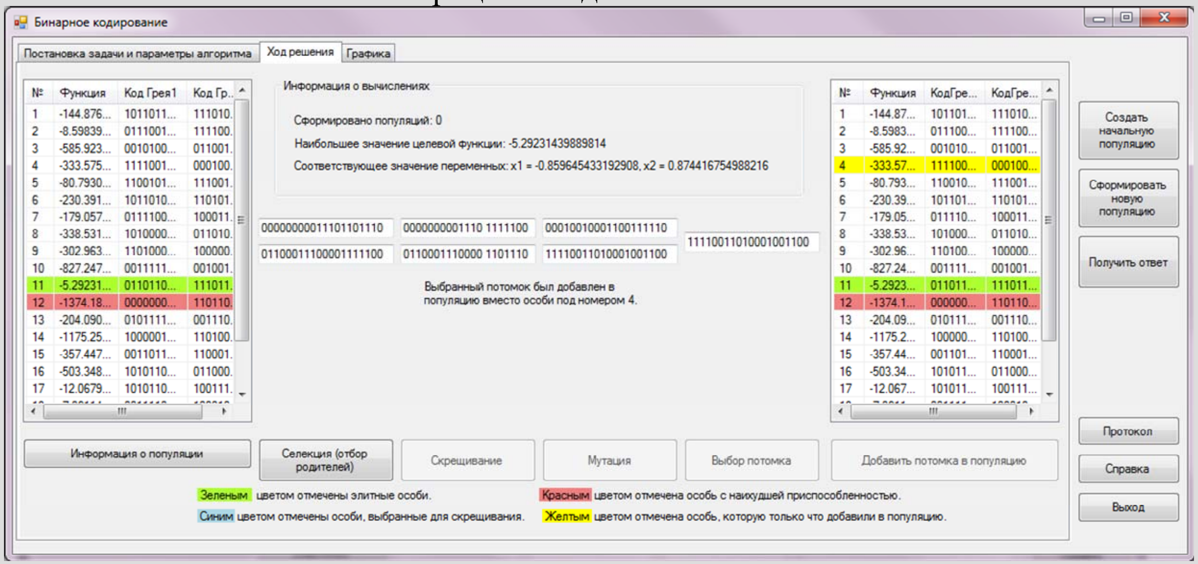
Анализ начальной популяции



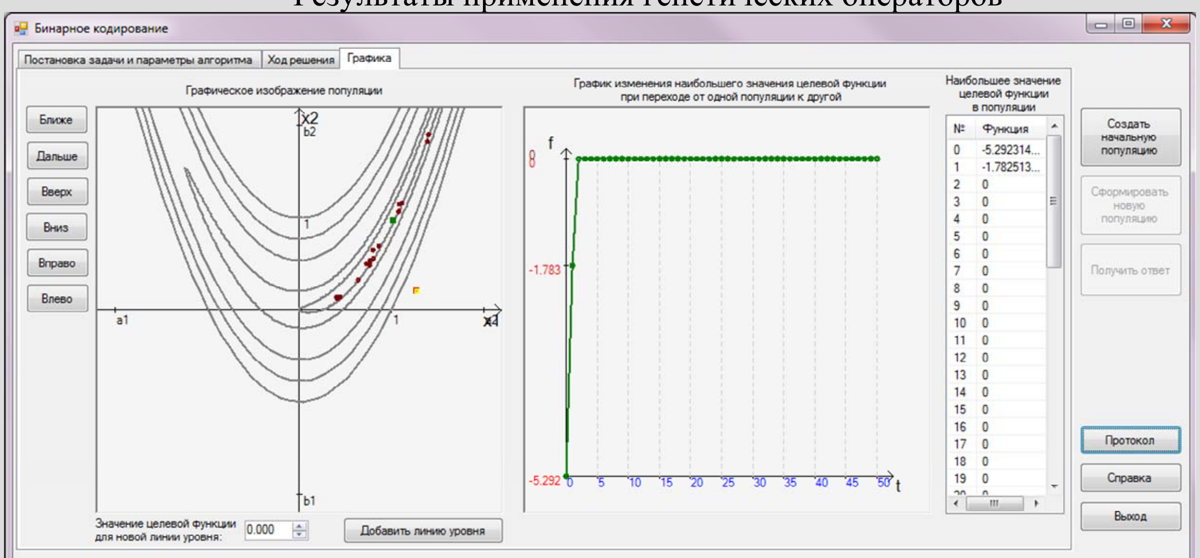


Родительские особи	Результат скрещивания	Результат мутации	Выбранный потомок
0000000001110 1101110	0000000001110 1111100	0001001000110 0111110	1111001101000 1001100
0110001110000 1111100	0110001110000 1101110	1111001101000 1001100	

Процесс создания потомка

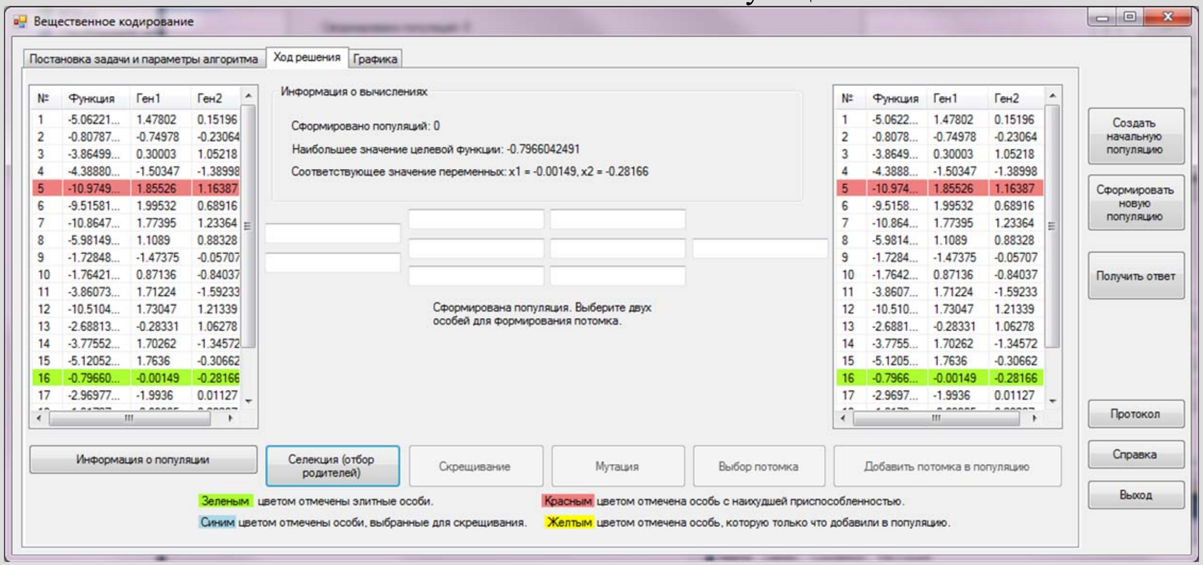


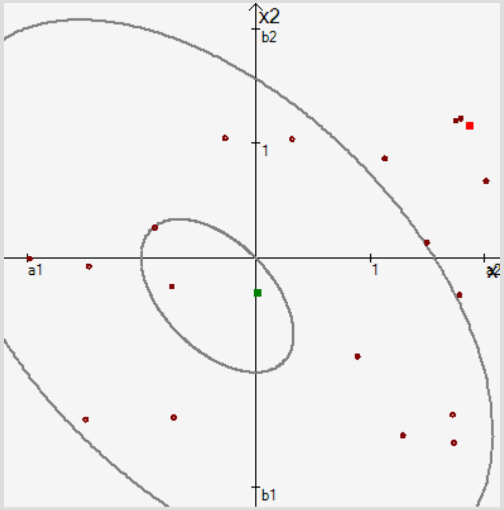
Результаты применения генетических операторов

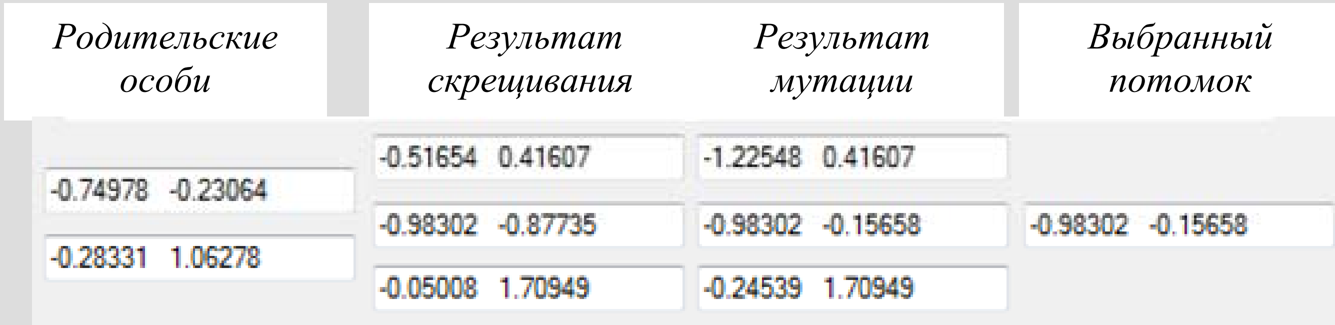


Пример. Покажем работу ***генетического алгоритма с вещественным кодированием*** на примере поиска условного максимума параболической функции (табл. П.1). Зададим параметры алгоритма: множество допустимых решений $-2 \le x_1 \le 2$, $-2 \le x_2 \le 2$, метод селекции – панмиксия, количество элитных особей – 1, метод скрещивания – линейный кроссовер, метод мутации – случайная, размер популяции $m = 20$ и количество популяций – $t_{\max} = 50$, – и создадим начальную популяцию.

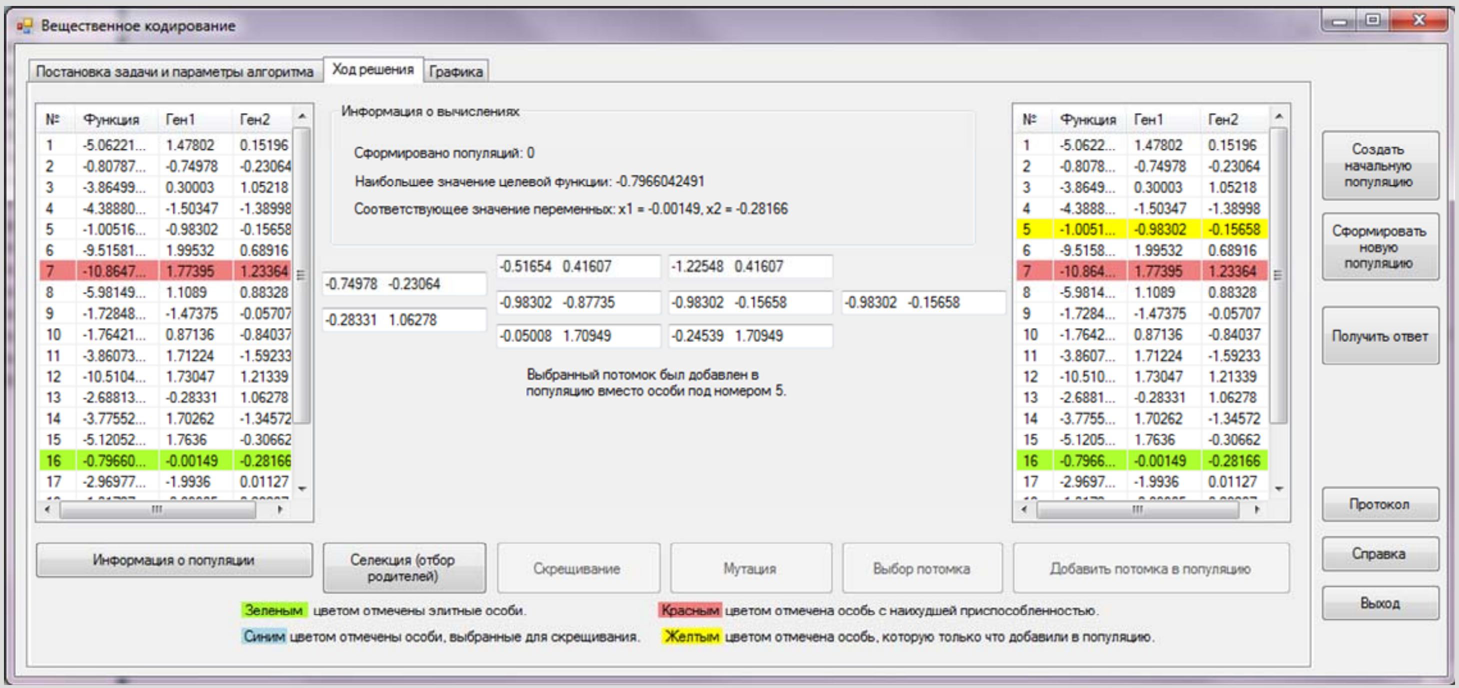
Анализ начальной популяции







Процесс выбора потомка



Результаты применения генетических операторов

Приближенное решение: $f = -0,667$, $x_1 = -0,3265$, $x_2 = -0,3346$. Точное решение задачи: $f^* = -2/3$, $x_1 = -1/3$, $x_2 = -1/3$.

Таблица

Целевая функция	ГА с бинарным кодированием				ГА с вещественным кодированием		
	l / p	Количество успехов	$\hat{E}[\Delta]$	$\hat{S}[\Delta]$	Количество успехов	$\hat{E}[\Delta]$	$\hat{S}[\Delta]$
Параболическая функция	14/0,1	1000	$3{,}04 \cdot 10^{-8}$	$9{,}92 \cdot 10^{-17}$	1000	$2{,}08 \cdot 10^{-6}$	$6{,}01 \cdot 10^{-12}$
Функция Розенброка	6/0,15	946	$5{,}51 \cdot 10^{-4}$	$1{,}73 \cdot 10^{-5}$	98 (265)	$1{,}51 \cdot 10^{-2}$ $(3{,}67 \cdot 10^{-3})$	$8\ 59 \cdot 10^{-4}$ $(9{,}22 \cdot 10^{-5})$
Функция Швефеля	20/0,05	999	$1{,}53 \cdot 10^{-7}$	$1{,}23 \cdot 10^{-11}$	5 (978)	$3{,}3 \cdot 10^{-3}$ $(2{,}5 \cdot 10^{-6})$	$2{,}49 \cdot 10^{-5}$ $(1{,}11 \cdot 10^{-11})$
Мульти-фукция	16/0,1	1000	$3{,}17 \cdot 10^{-7}$	$9{,}96 \cdot 10^{-14}$	1000	$1{,}25 \cdot 10^{-3}$	$2{,}09 \cdot 10^{-6}$
Корневая функция	14/0,1	1000	$5{,}98 \cdot 10^{-4}$	$2{,}49 \cdot 10^{-9}$	1000	$1{,}79 \cdot 10^{-2}$	$9{,}8 \cdot 10^{-5}$
Функция Шаффера	14/0,1	15	$4{,}67 \cdot 10^{-3}$	$1{,}58 \cdot 10^{-6}$	48	$3{,}47 \cdot 10^{-3}$	$3{,}82 \cdot 10^{-6}$
Функция Экли	16/0,1	1000	$1{,}94 \cdot 10^{-3}$	$1{,}46 \cdot 10^{-6}$	1000	$3{,}36 \cdot 10^{-3}$	$4{,}1 \cdot 10^{-6}$
Функция Растригина	14/0,1	1000	$6{,}68 \cdot 10^{-5}$	$2{,}46\ 10^{-8}$	1000	$3{,}8 \cdot 10^{-4}$	$1{,}9 \cdot 10^{-7}$

Таблица П.1

Название функции	**Формула**	**Множество допустимых решений**	**Глобальный экстремум** $f(x^*; y^*)$	**Точки глобального экстремума** $(x^*; y^*)^T$
Параболическая функция (рис. П.1,а, П.1,б)	$f(x,y) = -x^2 - y^2$	$x, y \in [-2; 2]$	0	$(0;0)^T$
	$f(x,y) = -2x^2 - xy - y^2 + 3x$		$-1,2857$	$(0,8571; -0,4285)^T$
Функция Розенброка (рис. П.2)	$f(x,y) = -a\left(y - x^2\right)^2 - \left(1-x\right)^2$	$x, y \in [-2; 2]$	0	$(1;1)^T$
Синусоидальная функция Швефеля (рис. П.3)	$f(x,y) = x\sin(\sqrt{\lvert x\rvert}) + y\sin(\sqrt{\lvert y\rvert})$	$x, y \in [-500; 500]$	837,9657	$(420,9687; 420,9687)^T$
Мульти-функция (рис. П.4)	$f(x,y) = x\sin(4\pi x) + y\sin(4\pi y)$	$x, y \in [-2; 2]$	4,2539	$(-1,6288; -1,6288)^T$ $(1,6288; 1,6288)^T$ $(-1,6288; 1,6288)^T$ $(1,6288; -1,6288)^T$
Корневая функция (рис. П.5)	$f(z) = \dfrac{1}{1+\left\lvert z^6 - 1\right\rvert}, z \in C, z = x + iy$	$x, y \in [-2; 2]$	1	$(0,5; -0,866)^T$ $(-0,5; 0,866)^T$ $(0,5; 0,866)^T$ $(-0,5; -0,866)^T$ $(1;0)^T$ $(-1;0)^T$
Функция Шаффера (рис. П.6)	$f(x,y) = \dfrac{1}{2} - \dfrac{\sin^2(\sqrt{x^2+y^2}) - 0.5}{(1 + 0.001(x^2+y^2))}$	$x, y \in [-10; 10]$	1	$(0;0)^T$
Функция Растригина (рис. П.7)	$f(x,y) = -20 + (10\cos(2\pi x) - x^2) +$ $+(10\cos(2\pi y) - y^2)$	$x, y \in [-5; 5]$	0	$(0;0)^T$

Продолжение таблицы П.1

Название функции	Формула	Множество допустимых решений	Глобальный экстремум $f(x^*; y^*)$	Точки глобального экстремума $(x^*; y^*)^T$
Трехгорбая функция (рис. П.8)	$f(x,y)=-2x^2+1,05x^4-\frac{1}{6}x^6-xy-y^2$	$x,y\in[-5;5]$	0	$(0;0)^T$
Функция Экли (рис. П.9)	$f(x,y)=-e+20\exp\left(-\sqrt{\frac{x^2+y^2}{50}}\right)+$ $+\exp\left(\frac{1}{2}(\cos(2\pi\,x)+\cos(2\pi\,y))\right)$	$x,y\in[-10;10]$	20	$(0;0)^T$
Функция «Птица» (рис П.10)	$f(x,y)=-\sin x\exp\left((1-\cos y)^2\right)-$ $-\cos y\exp\left((1-\sin x)^2\right)-(x-y)^2$	$x,y\in[-2\pi;2\pi]$	$106,7645$	$(4,7124;3,1416)^T$ $(-1,5708;-3,1416)^T$
Функция Букина 6 (рис. П.11)	$f(x,y)=-100\sqrt{\left\lvert -y+0,01x^2\right\rvert}-0,01\lvert x+10\rvert$	$x\in[-15;5]$, $y\in[-3;3]$	0	$(-10;1)^T$
Функция Швефеля 2.22 (рис. П.12)	$f(x,y)=-\lvert x\rvert-\lvert y\rvert-\lvert x\rvert\lvert y\rvert$	$x,y\in[-10;10]$	0	$(0;0)^T$
Функция Швефеля 1.2 (рис. П.13)	$f(x,y)=-x^2-(x+y)^2$	$x,y\in[-10;10]$	0	$(0;0)^T$
Двухэкстремальная функция (рис. П.14)	$f(x,y)=-3x^2-4y^2-23\cos(x-0,5)$	$x,y\in[-6;6]$	6,4892	$(-2,0709;0)^T$
Функция Griванка (рис П.15)	$f(x,y)=-1-\frac{x^2+y^2}{4000}+\cos\left(\frac{x}{\sqrt{1}}\right)\cos\left(\frac{y}{\sqrt{2}}\right)$	$x,y\in[-600;600]$	0	$(0;0)^T$

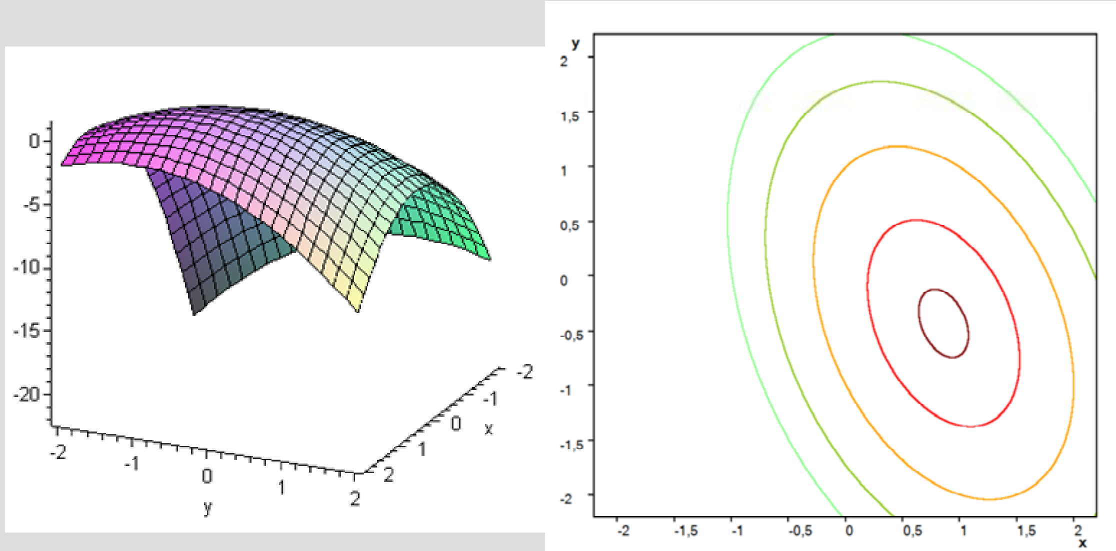


Рис. П.1, б. Изображения поверхности и линий уровня параболической функции

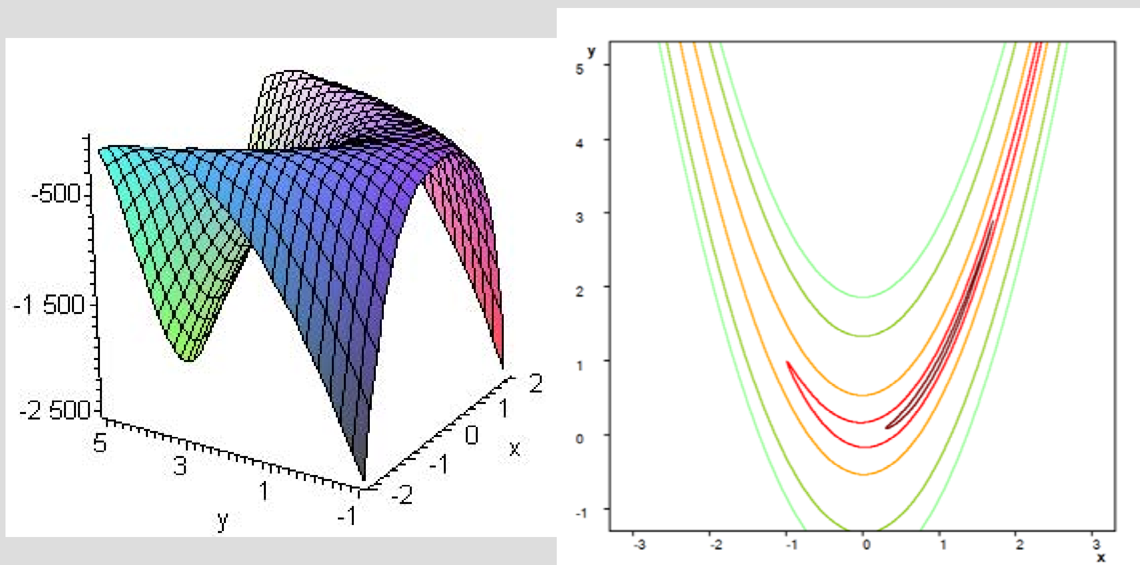


Рис. П.2. Изображения поверхности и линий уровня функции Розенброка при $a = 100$

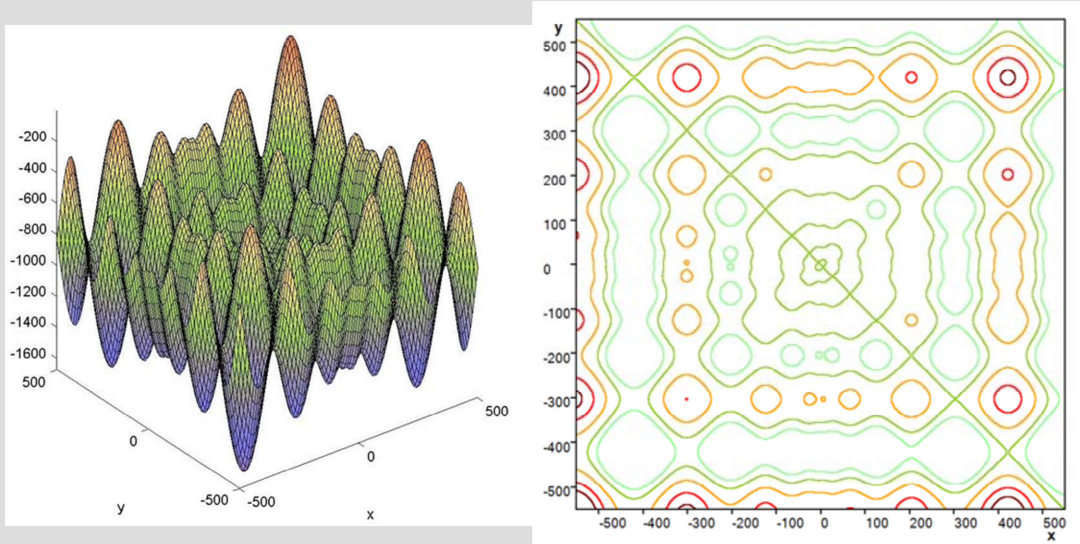


Рис. П.3. Изображения поверхности и линий уровня синусоидальной функции Швефеля

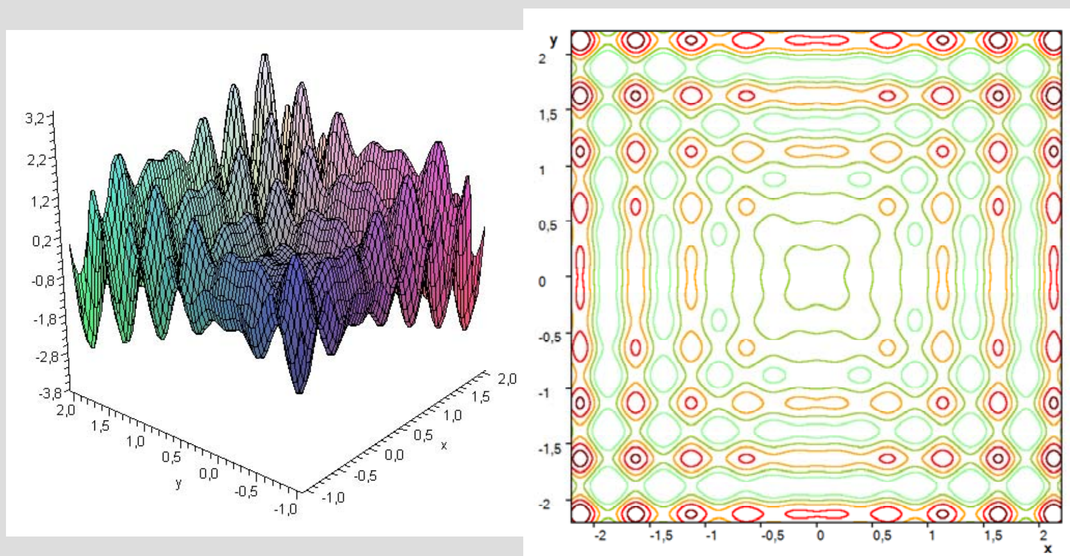


Рис. П.4. Изображения поверхности и линий уровня мульти-функции

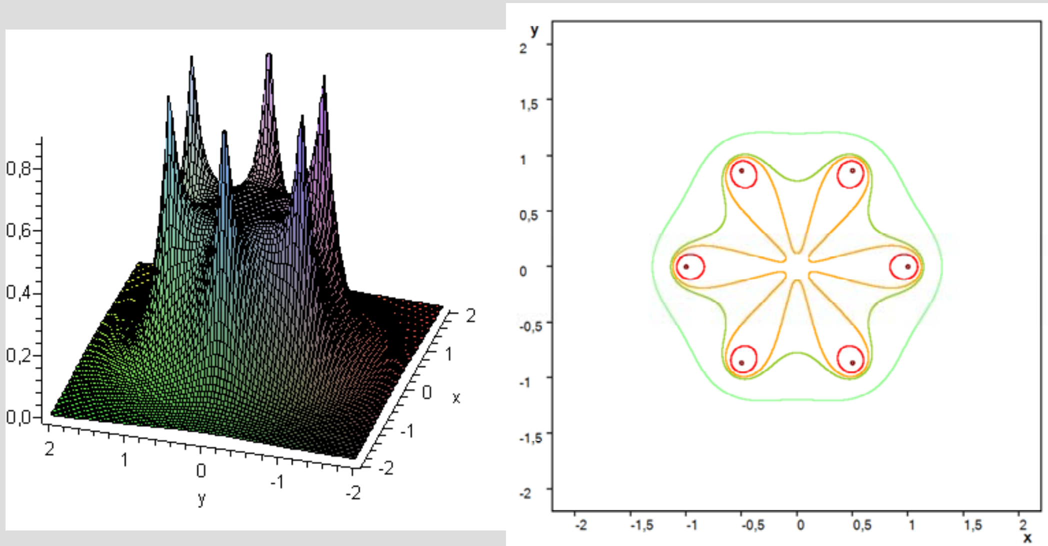


Рис. П.5. Изображения поверхности и линий уровня корневой функции

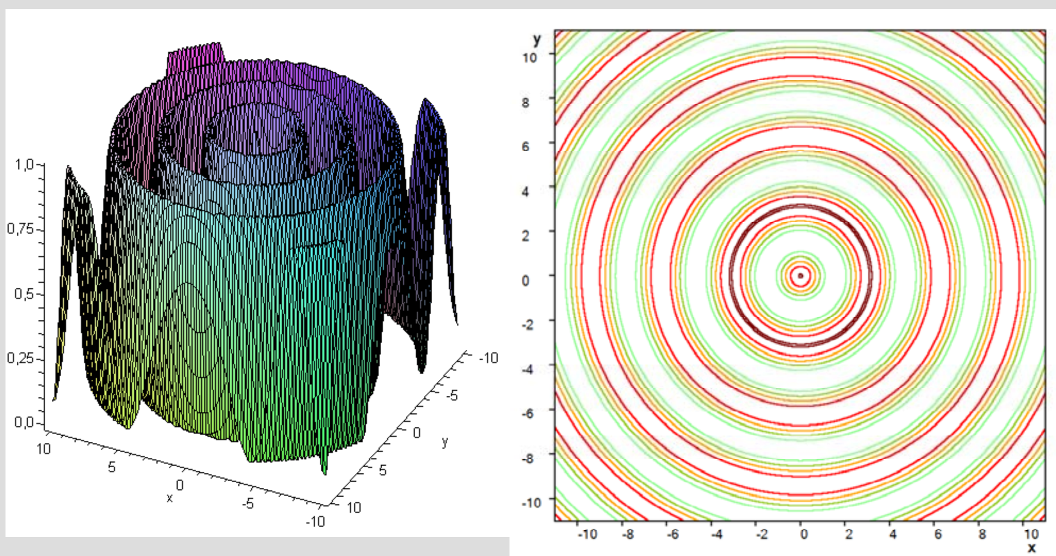


Рис. П.6. Изображения поверхности и линий уровня функции Шаффера

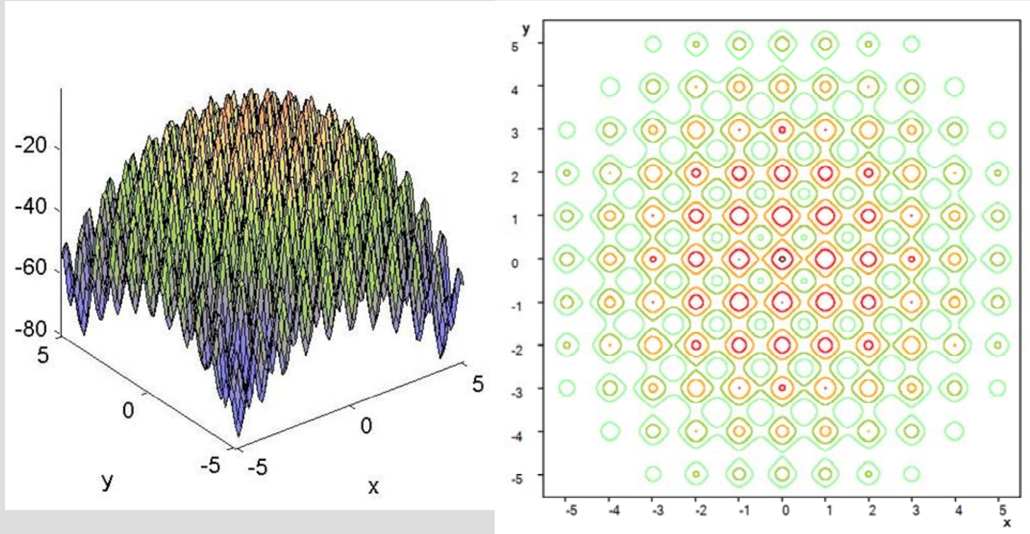


Рис. П.7. Изображения поверхности и линий уровня функции Растригина

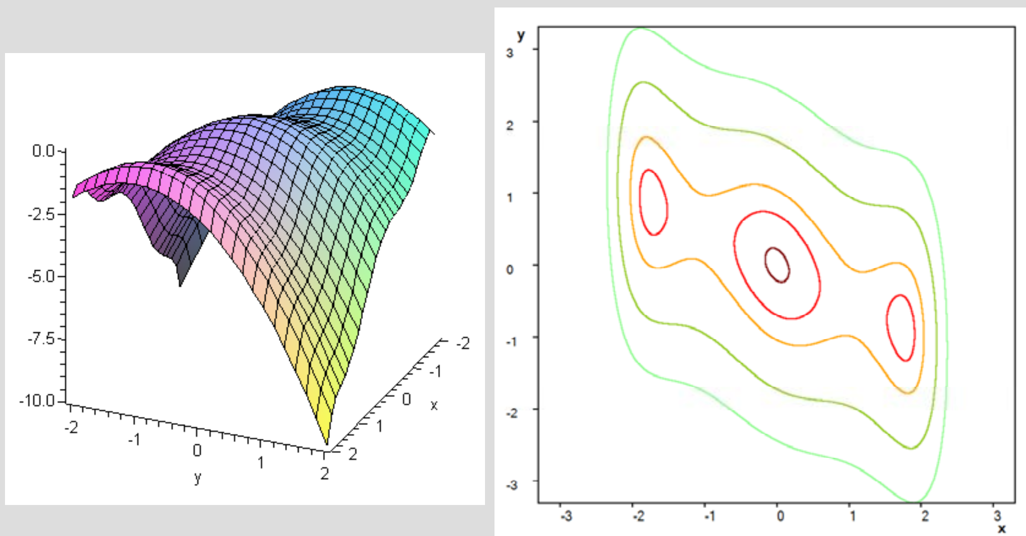


Рис. П.8. Изображения поверхности и линий уровня трехгорбой функции

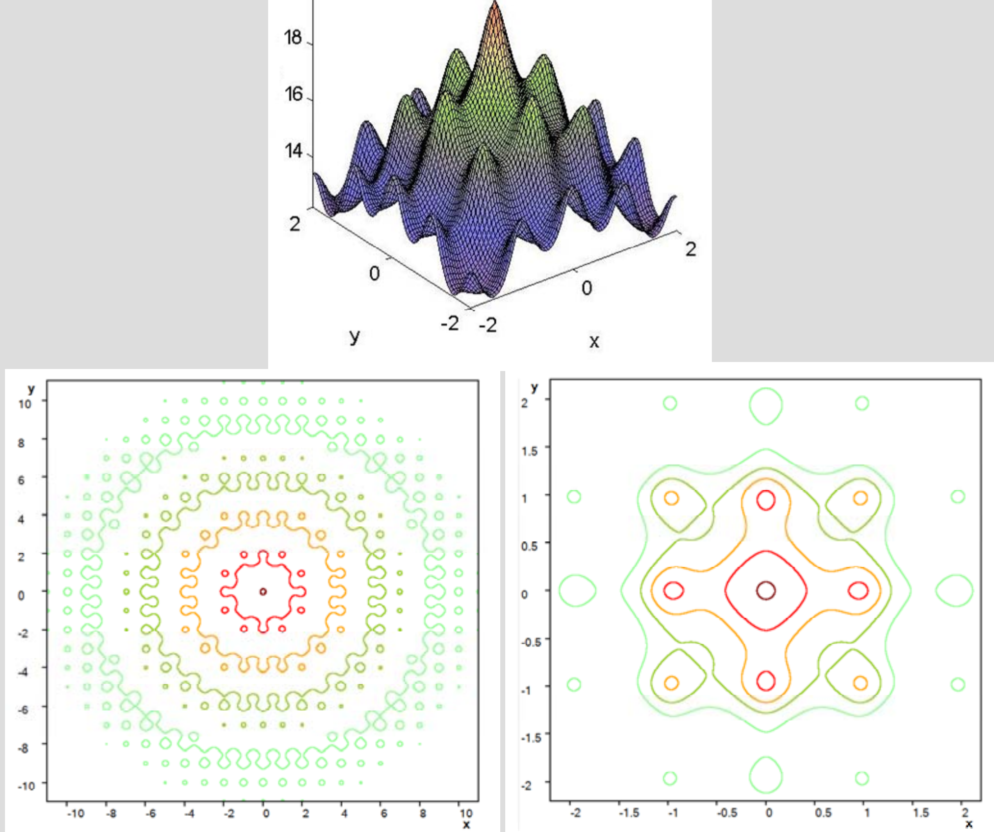


Рис. П.9. Изображения поверхности и линий уровня функции Экли

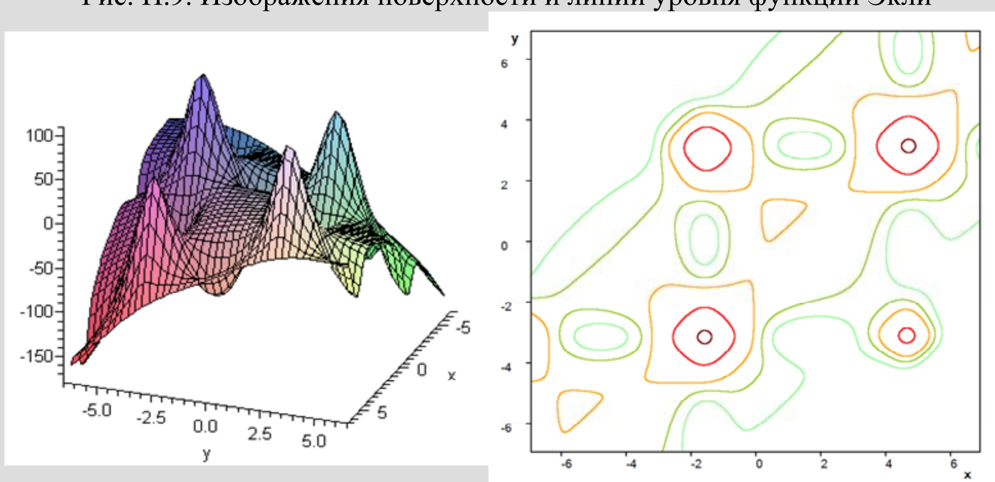


Рис. П.10. Изображения поверхности и линий уровня функции «Птица»

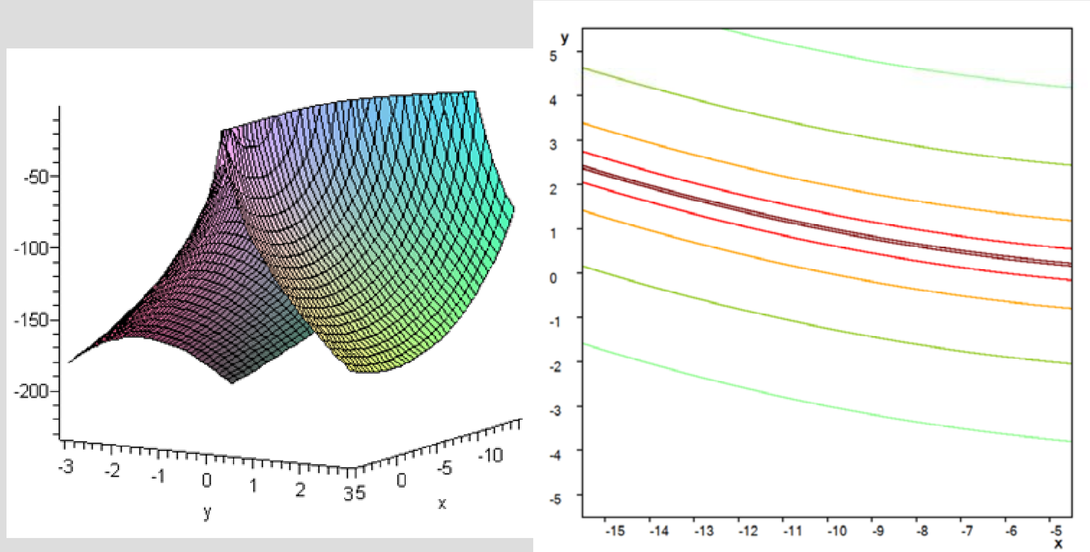


Рис. П.11. Изображения поверхности и линий уровня функции Букина 6

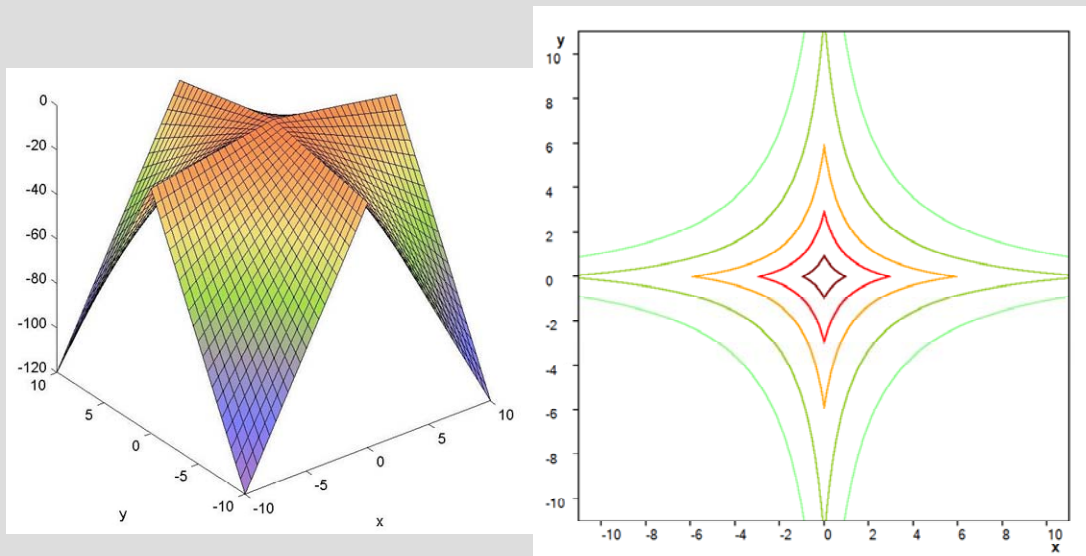


Рис. П.12. Изображения поверхности и линий уровня функции Швефеля 2.22

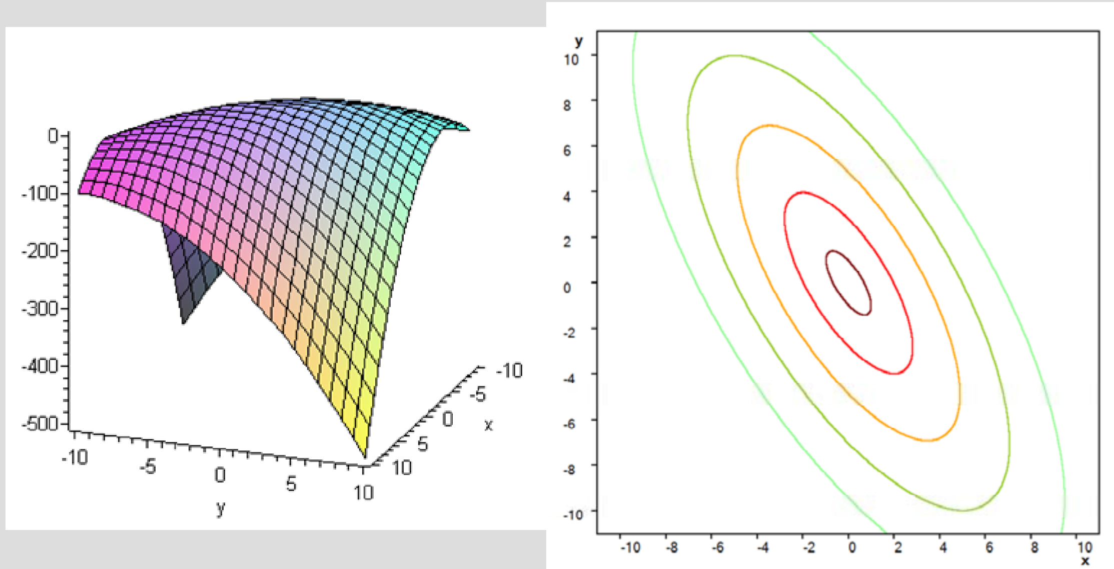


Рис. П.13. Изображения поверхности и линий уровня функции Швефеля 1.2

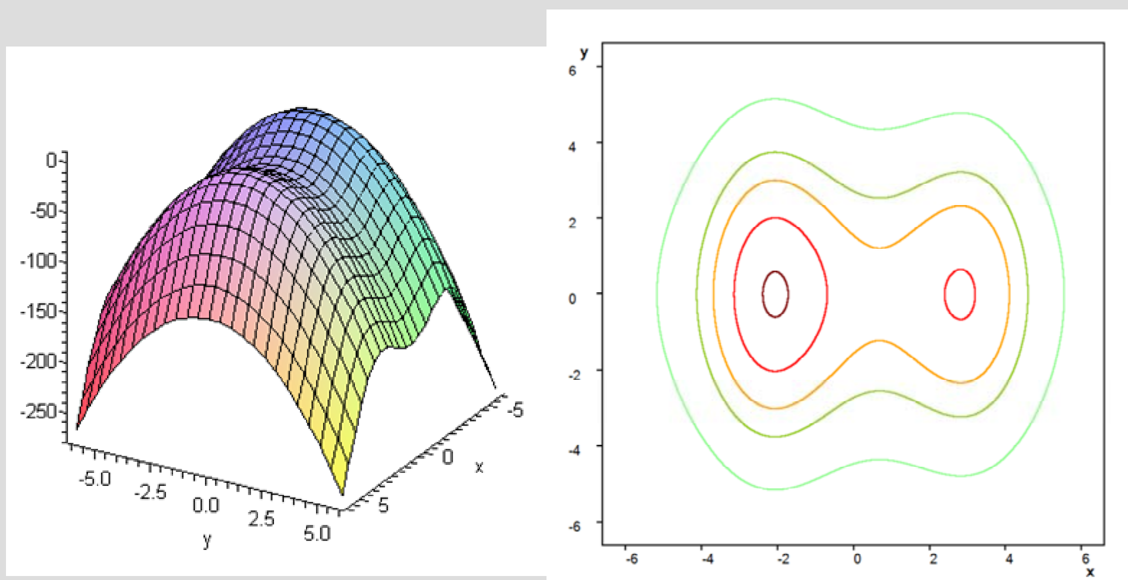


Рис. П.14. Изображения поверхности и линий уровня двухэкстремальной функции

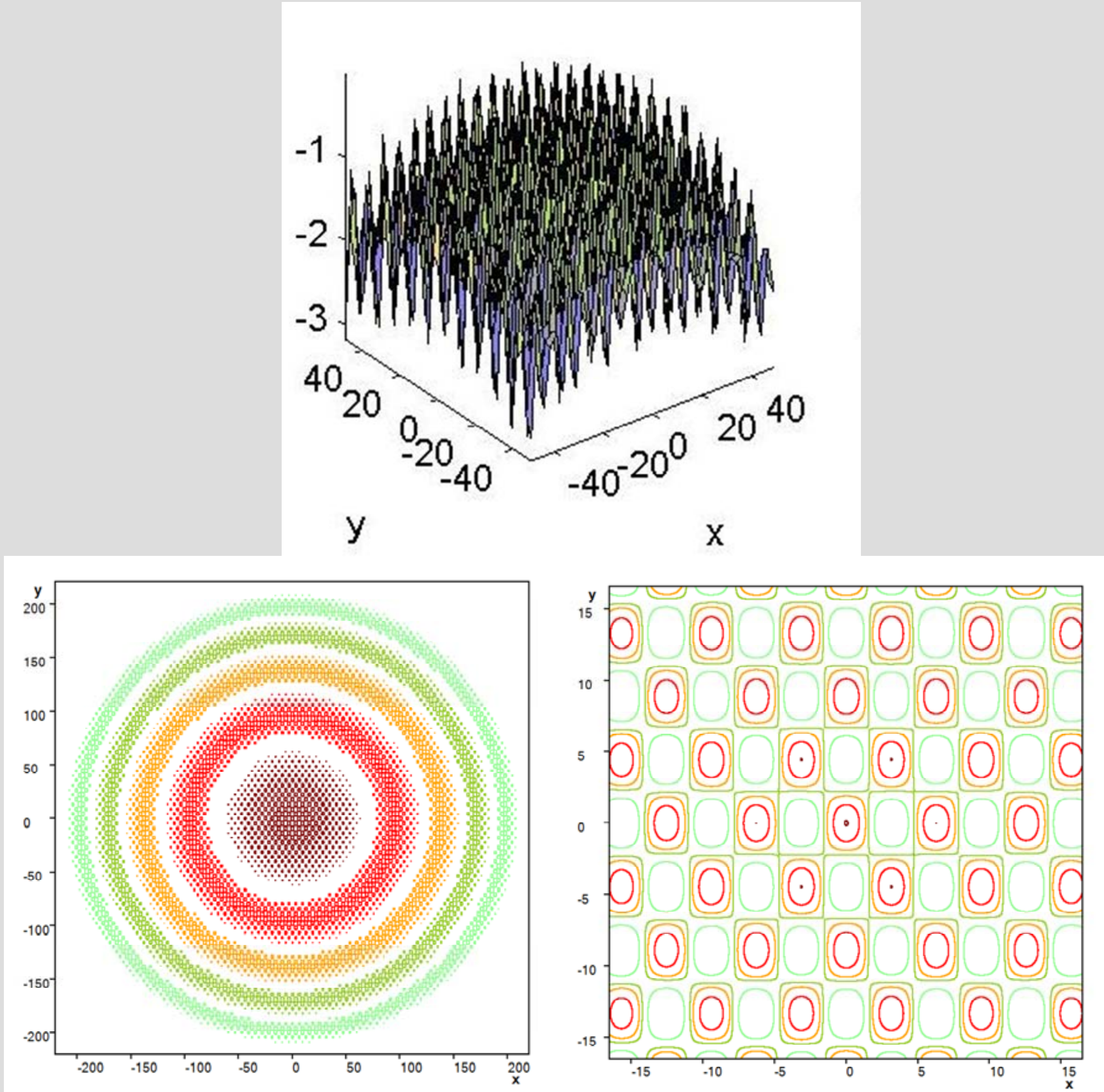


Рис. П.15. Изображения поверхности и линий уровня функции Griewank

I. Б. МЕТОДЫ, ИМИТИРУЮЩИЕ ИММУННЫЕ СИСТЕМЫ ОРГАНИЗМОВ

МЕТОД ИСКУССТВЕННЫХ ИММУННЫХ СИСТЕМ

Постановка задачи

Требуется найти глобальный условный минимум целевой функции $f(x)$ на множестве D, т.е. такую точку $x^* \in D$, что $f(x^*) = \min\limits_{x \in D} f(x)$, где $x = (x_1, x_2, ..., x_n)^T$, $D = \{x \mid x_i \in [a_i, b_i], i = 1, 2, ..., n\}$.

Стратегия поиска решения

Метод *искусственных иммунных систем* (ИИС) использует идеи, заимствованные из иммунологии, имитируя работу иммунной системы живого организма.

Иммунной системой живого организма называется подсистема, объединяющая органы и ткани, которые защищают организм от заболеваний. Назначение иммунной системы живого организма заключается в том, что она идентифицирует и уничтожает чужеродные тела, попавшие в организм, и совершенствуется, накапливая опыт борьбы с ними.

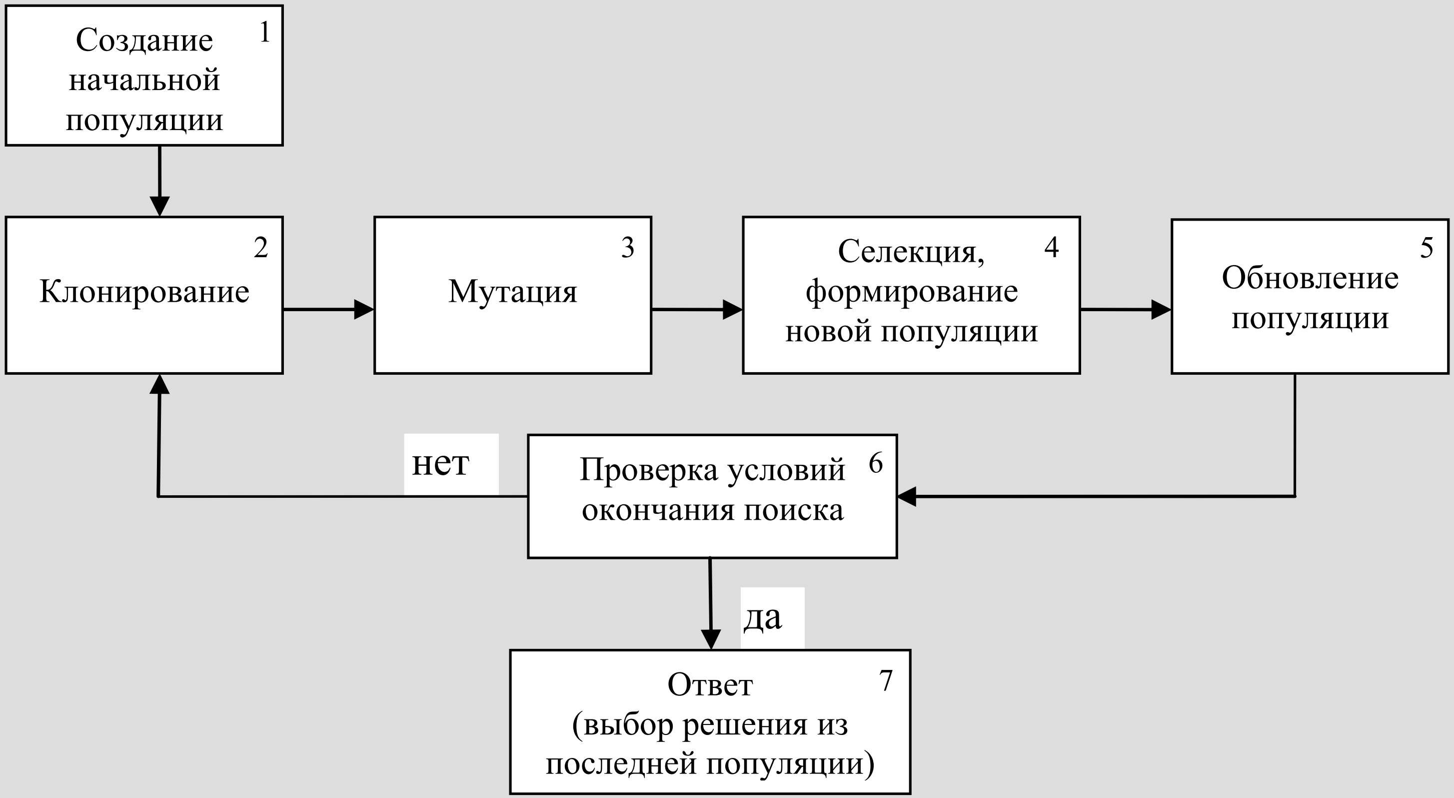
Антигеном называется вещество, которое воспринимается живым организмом как чужеродное и от которого организм пытается защититься. Для того чтобы организм смог защититься от антигенов, в нем при помощи специальных *иммунных клеток* вырабатываются антитела.

Антителом называется вещество, которое распознает антиген и способствует его уничтожению. Если иммунные клетки выработали антитела, которые смогли распознать антиген, то информация об этих антителах сохраняется в клетках памяти.

Клеткой памяти называется иммунная клетка, которая сохраняет в себе информацию о новых антителах, способных распознать антиген, для того, чтобы в следующий раз, когда в организм попадет такой же или похожий антиген, иммунная система смогла работать эффективнее.

Целевая функция $f(x)$ эквивалентна природному понятию приспособленности иммунной клетки к борьбе с антигенами, т.е. способности клетки вырабатывать антитела. Поэтому будем называть целевую функцию $f(x)$ *функцией приспособленности*. Вектор параметров $x = (x_1, x_2, ..., x_n)^T \in D$ целевой функции называется *иммунной клеткой*, которая вырабатывает антитела.

При решении задачи глобальной оптимизации используются конечные наборы $I = \{x^j = (x_1^j, x_2^j, \ldots, x_n^j)^T, j = 1, 2, \ldots, Np\} \subset D$ возможных решений, называемые *популяциями*, где x^j – иммунная клетка с номером j, Np – *размер популяции*. Чем меньше значение целевой функции $f(x^j)$, тем более иммунная клетка x^j приспособлена, т.е. способна вырабатывать необходимые антитела, и подходит в качестве решения.



Общая схема работы метода ИИС

Алгоритм решения задачи

Шаг 1. *Создание начальной популяции.*

Шаг 1.1. Задать количество иммунных клеток в популяции Np, число родительских клеток, выбираемых из популяции в ходе селекции s, число клеток в популяции с наихудшим значением функции приспособленности d, максимальное количество итераций K, параметр операции клонирования β или Nc в зависимости от варианта клонирования, параметр мутации r.

Шаг 1.2. Сгенерировать Np клеток начальной популяции на множестве D, используя равномерный закон распределения. Положить $k=0$ (счетчик итераций).

Шаг 1.3. Для всех клеток в популяции найти значение их функции приспособленности $f(x^j)$, $j=1,\ldots,Np$.

Результатом шага 1 является сформированная начальная популяция $I=\{x^j=(x_1^j,x_2^j,\ldots,x_n^j)^T,\ j=1,2,\ldots,Np\}\subset D$.

Шаг 2. *Клонирование.*

Шаг 2.1. Упорядочить клетки в популяции по возрастанию соответствующего им значения функции приспособленности: $f(x^{(1)}),\ldots,f(x^{(Np)})$, где $f(x^{(1)})=f_{\min}$, $f(x^{(Np)})=f_{\max}$. Выбрать из упорядоченной популяции s родительских клеток с наилучшими соответствующими значениями функции приспособленности (s первых клеток).

Шаг 2.2. Для каждой выбранной клетки используют два варианта клонирования:

1) «пропорциональное» – для каждой клетки с номером j в упорядоченной популяции генерируется N_j клонов, где $N_j=\left[\dfrac{\beta\cdot Np}{j}\right]$, $j=1,\ldots,s$; β – параметр операции клонирования;

2) «равномерное» – для каждой клетки генерируются N_j клонов, где $N_j=Nc$, а Nc – параметр операции клонирования.

Результатом шага 2 является популяция, состоящая из Np первоначальных клеток и клонов s родительских клеток, т.е. всего $Np+\sum_{j=1}^{s}N_j$ клеток.

Шаг 3. *Мутация.*

Для каждой родительской клетки с номером j выполнить процедуру мутации для всех ее N_j клонов. Для этого каждую координату $x_i^{c,j}$ клона $x^{c,j}$ заменить на $y_i^{c,j}$:

1) используя равномерный закон распределения на отрезке [0;1], сгенерировать число u;

2) если $u > 0,5$, то положить $y_i^{c,j} = x_i^{c,j} + U(0; b_i - x_i^j) \cdot r$;

если $u \le 0,5$, то $y_i^{c,j} = x_i^{c,j} - U(0; x_i^j - a_i) \cdot r$,

где $j = 1,...,s$, $c = 1,...,N_j$, $i = 1,...,n$, $U(a;b)$ – случайная величина, равномерно распределенная на отрезке $[a;b]$, r – параметр мутации;

3) если $y_i^{c,j} \notin [a_i; b_i]$, то процедуру повторить.

Результатом шага 3 является популяция, состоящая из Np первоначальных клеток и клонов-мутантов s родительских клеток, т.е. всего $Np + \sum_{j=1}^{s} N_j$ клеток.

Шаг 4. *Селекция, формирование новой популяции.*

Шаг 4.1. Вычислить значение функции приспособленности для каждого клона-мутанта $f(y^{c,j})$, $c = 1,...,N_j$, $j = 1,...,s$.

Шаг 4.2. Для каждой родительской клетки x^j, $j = 1,...,s$, среди ее клонов-мутантов найти клетку с наименьшим значением функции приспособленности $f_{\min}(y^j)$. Если $f_{\min}(y^j) < f(x^j)$, то заменить в новой популяции родительскую клетку клоном-мутантом y^j, а иначе – оставить родительскую клетку x^j.

Результатом шага 4 является новая популяция из Np клеток.

Шаг 5. *Обновление популяции.*

Шаг 5.1. Выбрать из популяции d клеток с наихудшим значением функции приспособленности (d последних клеток) и заменить их новыми клетками, генерируемыми случайным образом на множестве D.

Шаг 5.2. Вычислить значение функции приспособленности новых клеток.

Шаг 5.3. Положить $k = k + 1$.

Результатом шага 5 является обновленная популяция, состоящая из Np клеток.

Шаг 6. *Проверка условий окончания поиска.*

Если $k = K$, то поиск завершить, перейти к шагу 7.

Если $k < K$, то поиск продолжить, перейти к шагу 2.

Шаг 7. *Выбор решения из последней популяции.*

Закончить работу алгоритма. В качестве решения (приближенного) задачи $f(x^*) = \min\limits_{x \in D} f(x)$ выбрать клетку с наименьшим значением функции приспособленности из текущей популяции: $x^* \cong \tilde{x}^* = \arg\min\limits_{j=1,\ldots,Np} f(x^j)$.

РАСШИРЕННЫЙ МЕТОД ИСКУССТВЕННЫХ ИММУННЫХ СИСТЕМ

Постановка задачи

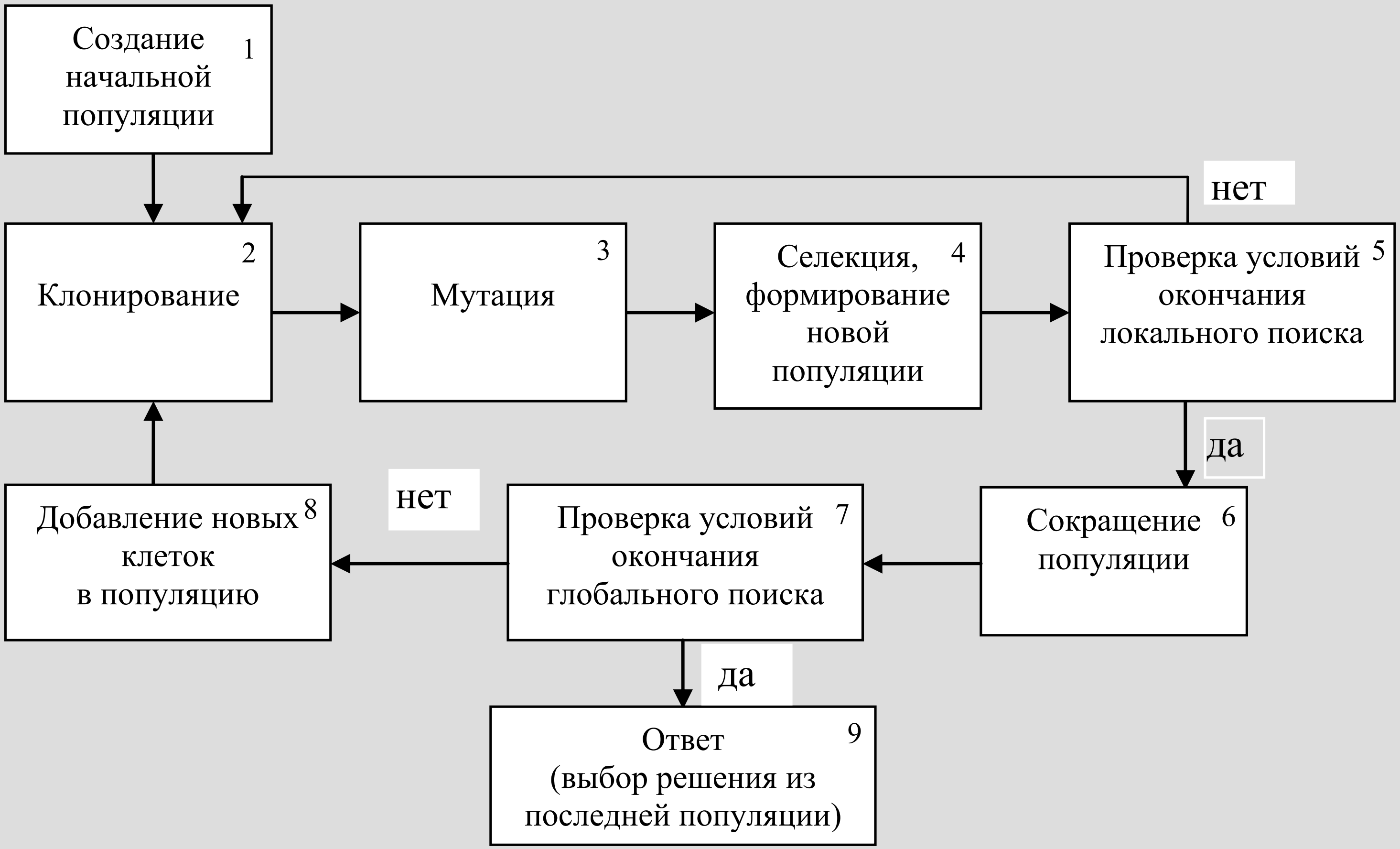
Требуется найти глобальный условный максимум целевой функции $f(x)$ на множестве D, т.е. такую точку $x^* \in D$, что $f(x^*) = \max\limits_{x \in D} f(x)$, где $x = (x_1, x_2, ..., x_n)^T$, $D = \{x \mid x_i \in [a_i, b_i], i = 1, 2, ..., n\}$.

Стратегия поиска решения

Особенностью расширенного метода ИИС является *сочетание локального и глобального поиска решения.*

Глобальный поиск реализуется в виде циклического итерационного процесса. Каждая итерация глобального поиска представляет собой локальный поиск. В конце каждой итерации происходит сокращение популяции с использованием идей *кластеризации*. Оставшиеся иммунные клетки помечаются как клетки памяти, после чего проверяются условия окончания работы метода. Если условия окончания не выполнены, то к популяции присоединяются новые особи, количество которых пропорционально размеру популяции. Таким образом, размер популяции Np в расширенном методе ИИС – переменная величина. Если условие окончания работы метода выполнено, то в качестве приближенного решения задачи из последней популяции выбираются иммунные клетки, которым соответствует наибольшее значение функции приспособленности.

Локальный поиск реализуется в виде циклического итерационного процесса, во время которого к популяции применяются биологические операторы: клонирование, мутация и селекция. Таким образом, происходит смена популяции на новую, к которой, если условие окончания локального поиска не выполнены, опять применяются биологические операторы, и т.д. до выполнения условия окончания локального поиска. Средняя приспособленность популяции при этом будет расти. Если условие окончания локального поиска выполнено, то начинается новая итерация глобального поиска.



Общая схема работы расширенного метода ИИС

Алгоритм решения задачи

Шаг 1. *Создание начальной популяции.*

Шаг 1.1. Задать количество иммунных клеток в начальной популяции Ip, порог близости между иммунными клетками σ, процент добавляемых в популяцию новых иммунных клеток d, максимальное количество итераций локального поиска K, порог близости между средними значениями функции приспособленности двух популяций ε, параметр операции клонирования β или Nc в зависимости от выбора варианта клонирования, параметр мутации γ.

Шаг 1.2. Сгенерировать Ip клеток начальной популяции на множестве D, используя равномерный закон распределения. Положить $k=0$ (счетчик итераций), $Np = Ip$ – количество клеток в текущей популяции, $m=0$ – счетчик итераций глобального поиска.

Шаг 1.3. Для всех клеток в популяции найти значение их функции приспособленности $f(x^j)$, $j=1,...,Np$ и среднюю приспособленность популяции $\overline{f}_k = \frac{1}{Np}\sum_{j=1}^{Np} f(x^j)$.

Результатом шага 1 является сформированная начальная популяция $I=\{x^j=(x_1^j,x_2^j,\ldots,x_n^j)^T,\ j=1,2,\ldots,Np\}\subset D$.

Шаг 2. *Клонирование.*

Шаг 2.1. Для всех элементов популяции найти нормализованное значение функции приспособленности (принадлежит отрезку [0;1]) $\hat{f}(x^j) = \frac{f(x^j)-f_{\min}}{f_{\max}-f_{\min}}$, $j=1,...,Np$, где $f_{\max}, f_{\min}$ – максимальное и минимальное значения функции приспособленности в популяции соответственно.

Шаг 2.2. Используют два варианта клонирования:

1) «пропорциональное» – клетки в популяции упорядочиваются по убыванию соответствующей им функции приспособленности: $f(x^{(1)}),...,f(x^{(Np)})$, где $f(x^{(1)}) = f_{\max}$, $f(x^{(Np)}) = f_{\min}$. Для каждой клетки с номером j в упорядоченной популяции генерируется N_j клонов, где $N_j = \left[\frac{\beta\cdot Np}{j}\right]$, $j=1,...,Np$, β – параметр операции клонирования;

2) «равномерное» – для каждой клетки популяции, называемой *родительской клеткой*, генерируются N_j клонов, где $N_j = Nc$, Nc – параметр операции клонирования.

Результатом шага 2 является популяция, состоящая из Np родительских клеток и их клонов, т.е. всего $Np + \sum_{j=1}^{Np} N_j$ клеток.

Шаг 3. *Мутация.*

Для каждой родительской клетки с номером j выполнить процедуру мутации для всех ее N_j клонов. Для этого каждую координату клона $x^{c,j}$ заменяют на $y_i^{c,j}$: $y_i^{c,j} = x_i^{c,j} + \alpha \cdot N(0;1)$, $j = 1,...,Np$, $c = 1,...,N_j$, $i = 1,...,n$, где $\alpha = \frac{1}{\gamma} e^{-\hat{f}(x_j)}$, где $N(0;1)$ – стандартная гауссовская случайная величина, γ – параметр мутации. Если $y_i^{c,j} \notin [a_i; b_i]$, то процедуру повторить.

Результатом шага 3 является популяция, состоящая из Np родительских клеток и их клонов-мутантов, т.е. всего $Np + \sum_{j=1}^{Np} N_j$ клеток.

Шаг 4. *Селекция, формирование новой популяции.*

Шаг 4.1. Вычислить значение функции приспособленности для каждого клона-мутанта $f(y^{c,j})$, $c = 1,...,N_j$, $j = 1,...,Np$.

Шаг 4.2. Для каждой родительской клетки x^j, $j = 1,...,Np$, среди ее клонов-мутантов найти клетку с наибольшим значением функции приспособленности $f_{\max}(y^j)$. Если $f_{\max}(y^j) > f(x^j)$, то поместить в новую популяцию клона-мутанта y^j, а иначе – родительскую клетку x^j.

Шаг 4.3. Вычислить среднюю приспособленность сформированной популяции $\overline{f}_k = \frac{1}{Np} \sum_{j=1}^{Np} f(x^j)$.

Результатом шага 4 является новая популяция из Np клеток.

Шаг 5. *Проверка условий окончания локального поиска.*

Шаг 5.1. Если $k = 0$, то перейти к шагу 2.

Если $k = K$, то перейти к шагу 9.

Если $0 < k < K$, то перейти к шагу 5.2.

Шаг 5.2. Если $\left| \overline{f}_k - \overline{f}_{k-1} \right| \le \varepsilon$, то закончить локальный поиск. Перейти к шагу 6.

Если $\left| \overline{f}_k - \overline{f}_{k-1} \right| > \varepsilon$, то продолжить локальный поиск. Положить $k = k + 1$ и перейти к шагу 2.

Шаг 6. *Сокращение популяции.*

Вычислить евклидово расстояние между иммунными клетками в популяции. Если евклидово расстояние между двумя клетками меньше заданной пороговой величины σ, то из двух клеток удаляется та, которой соответствует худшее значение функции приспособленности.

Шаг 6.1. Упорядочить клетки в популяции по убыванию значения их функции приспособленности $f(x^{(1)}),...,f(x^{(Np)})$, $f(x^{(1)}) = f_{\max}$. Положить $Ns = Np$ – количество клеток в сокращенной популяции.

Шаг 6.2. Положить $p = 0$.

Шаг 6.3. Положить $s = p + 1$.

Шаг 6.4. Если $\sqrt{\sum_{i=1}^{n}(x_i^{(p)} - x_i^{(s)})^2} > \sigma$, то перейти к шагу 6.5, иначе, если $\sqrt{\sum_{i=1}^{n}(x_i^{(p)} - x_i^{(s)})^2} \le \sigma$, то клетку $x^{(s)}$ удалить из популяции, положить $Ns = Ns - 1$, $s = s - 1$ и перейти к шагу 6.5.

Шаг 6.5. Если $s < Ns$, то положить $s = s + 1$, перейти к шагу 6.4.

Если $s = Ns$, то положить $p = p + 1$, перейти к шагу 6.6.

Шаг 6.6. Если $p < Ns$, то перейти к шагу 6.3.

Если $p = Ns$, то закончить сокращение популяции.

Результатом шага 6 является сокращенная популяция, состоящая из Ns клеток.

Шаг 7. *Проверка условий окончания глобального поиска.*

Шаг 7.1. Положить $Mp_m = Ns$ – количество клеток в сокращенной популяции на m-й итерации глобального поиска.

Шаг 7.2. Если $m = 0$, то положить $m = m + 1$, поиск продолжить, переходя к шагу 8.

Если $m > 0$, то перейти к шагу 7.3.

Шаг 7.3. Если $Mp_m = Mp_{m-1}$, то глобальный поиск закончить и перейти к шагу 9.

Если $Mp_m \neq Mp_{m-1}$, то глобальный поиск продолжить, переходя к шагу 8.

Шаг 8. *Добавление новых клеток в популяцию.*

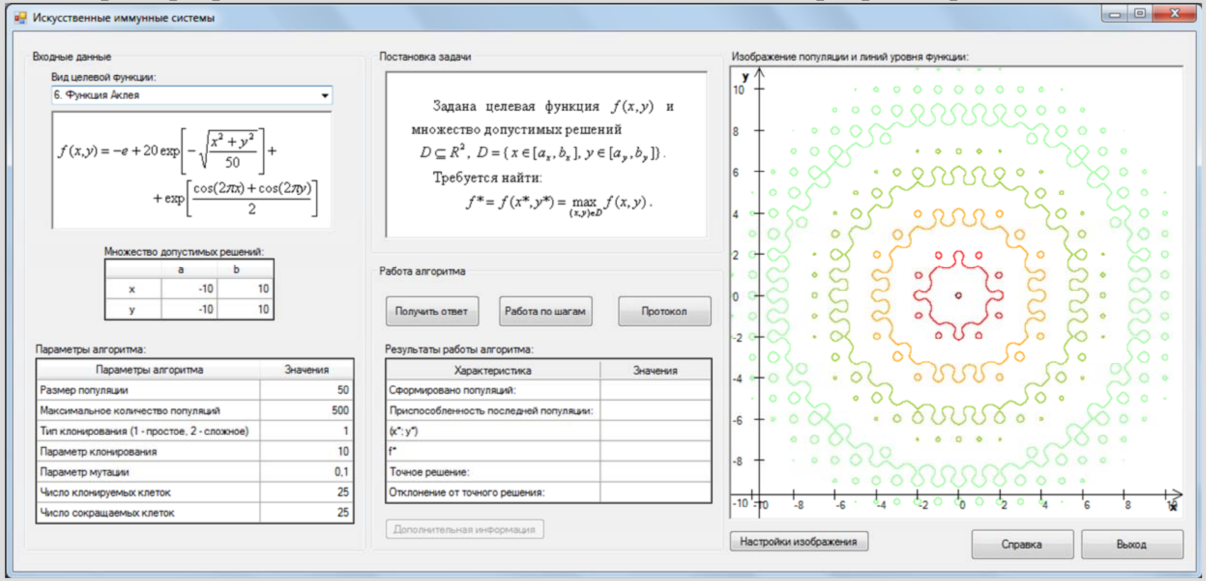
Шаг 8.1. Используя равномерный закон распределения на множестве D, добавить в популяцию $N_{new} = \left[\frac{Ns \cdot d}{100\%}\right]$ новых клеток. Положить $Np = Ns + N_{new}$, $k = k + 1$, перейти к шагу 2.

Результатом шага 8 является популяция, состоящая из $Np = Ns + N_{new}$ клеток.

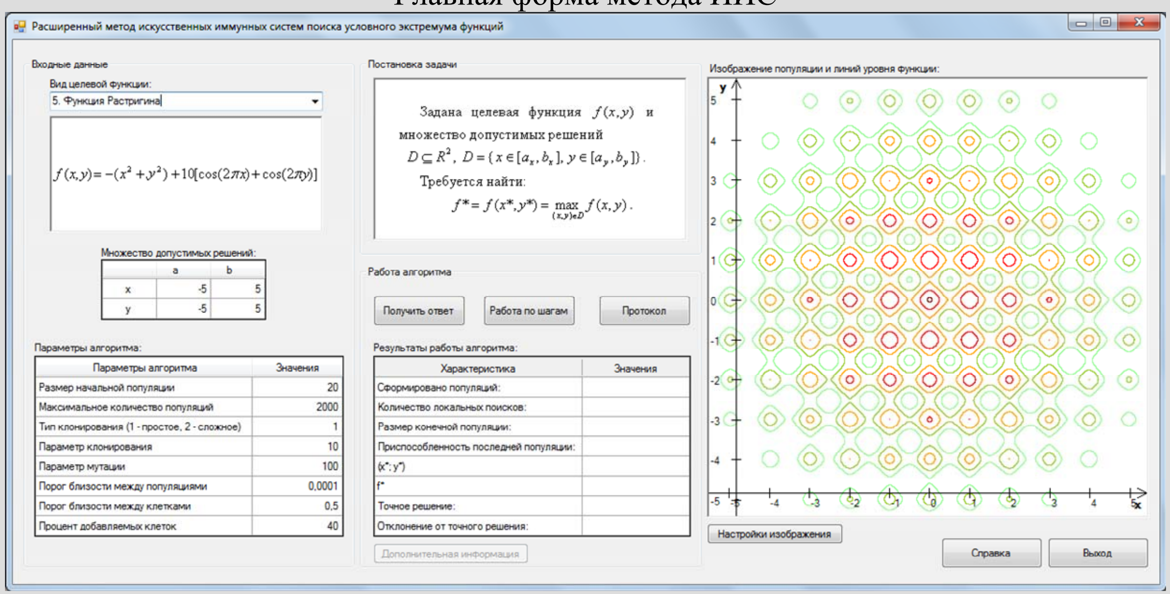
Шаг 9. *Выбор решения из последней популяции.*

Закончить работу алгоритма. В качестве решения (приближенного) задачи (2.2) выбрать клетку с наибольшим значением функции приспособленности из текущей популяции: $x^* \cong \tilde{x}^* = \arg\max_{j=1,...,Np} f(x^j)$.

Среда разработки Microsoft Visual Studio 2010, язык программирования C# .



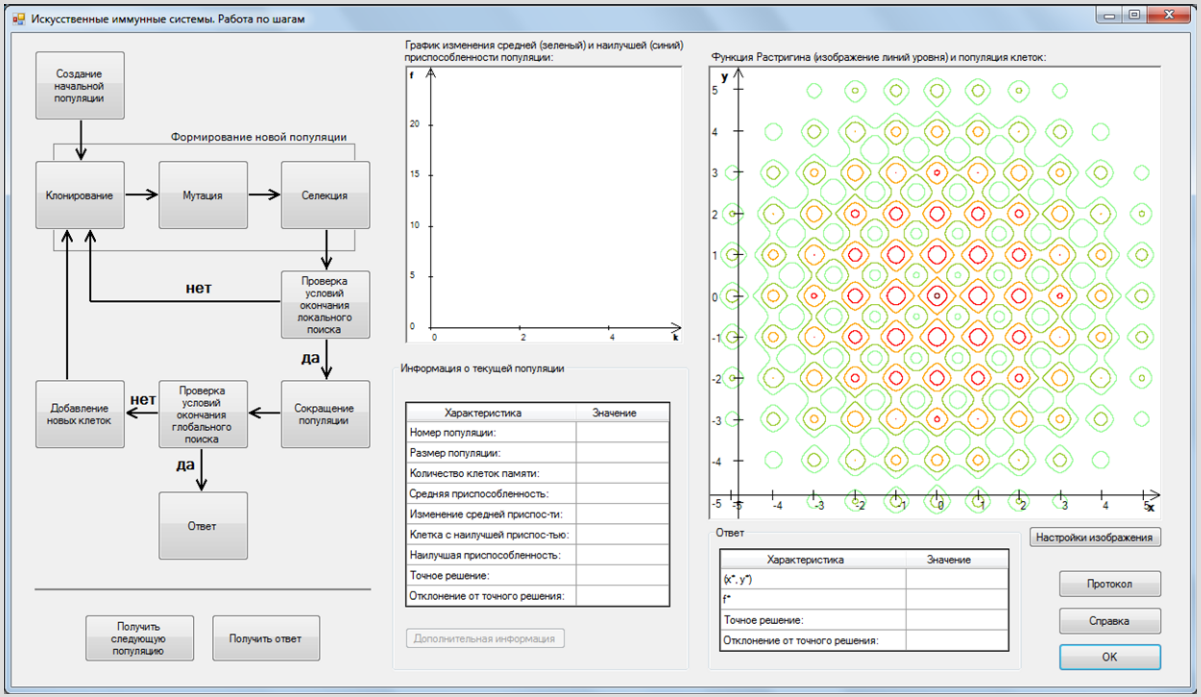
Главная форма метода ИИС



Главная форма расширенного метода ИИС

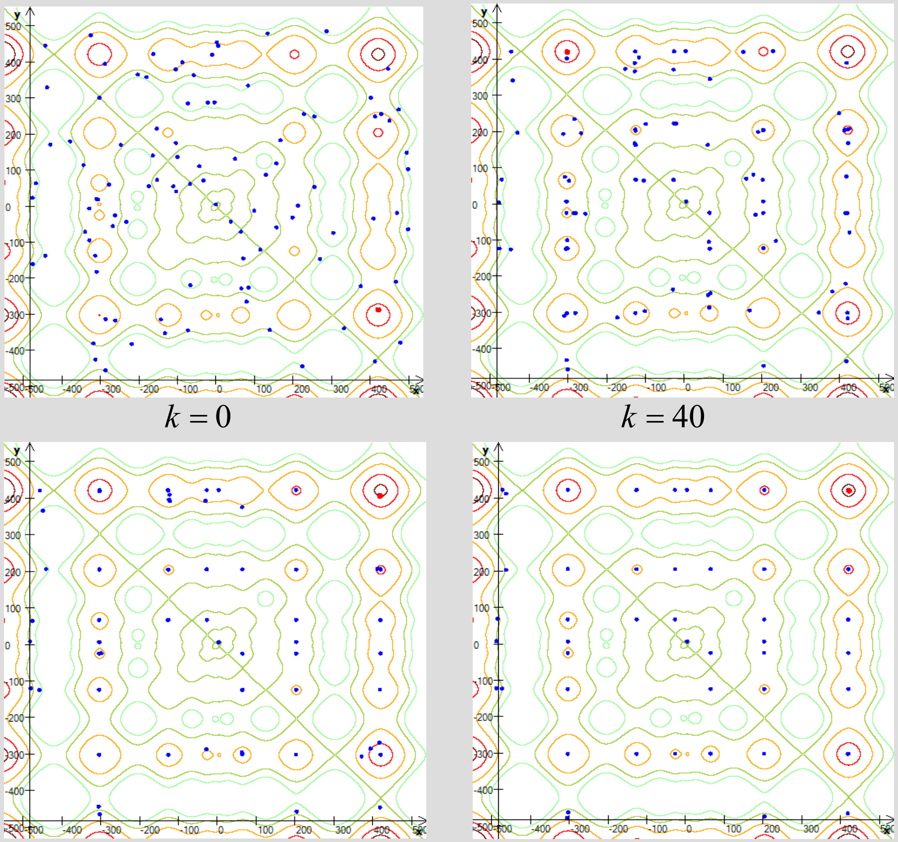
Применение метода искусственных иммунных систем

Пример 1. Рассмотрим функцию Швефеля (табл. П.1). Зададим множество допустимых решений $x, y \in [-500; 500]$. Выберем следующие параметры алгоритма: размер популяции $Np = 100$; максимальное количество популяций $K = 100$; тип клонирования – «равномерное»; количество клонируемых клеток $s = 100$; параметр клонирования $Nc = 10$; параметр мутации $r = 0,01$; количество удаляемых клеток с наихудшей приспособленностью $d = 0$.



Форма работы расширенного метода ИИС по шагам

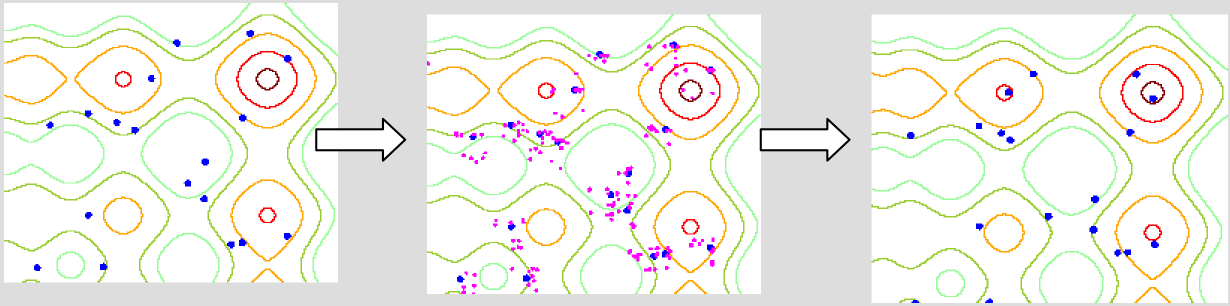
На рис. представлена начальная ($k = 0$), промежуточные ($k = 40, k = 50$) и конечная ($k = 100$) популяции клеток. Результаты работы метода ИИС: клетка с наилучшей приспособленностью $(x^*; y^*)^T = (421,0005; 421,0105)^T$; наилучшая приспособленность $f(x^*; y^*) = 837,9654$; отклонение от точного решения $\Delta = 0,0003$.



$k = 0$ $k = 40$

$k = 50$ $k = 100$

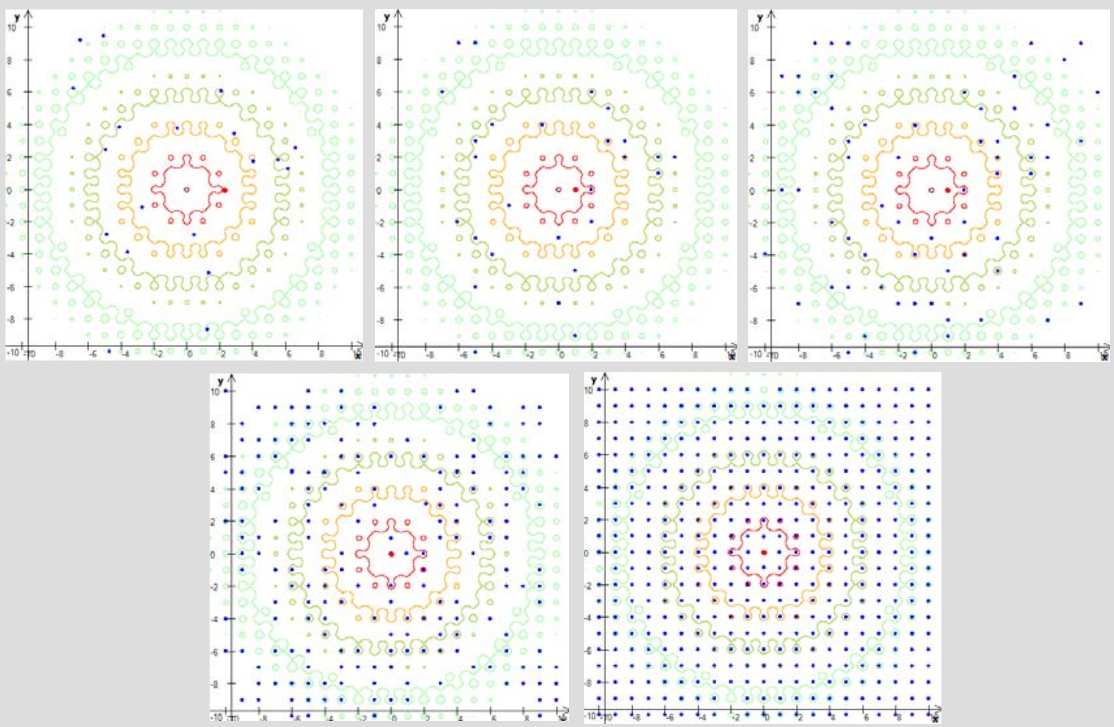
Начальная, промежуточные и конечная популяции ($s = 100$, $d = 0$)



Процесс перехода от одной популяции к другой

Пример 2. Рассмотрим функцию Экли (табл. П.1). Зададим множество допустимых решений $x, y \in [-10;10]$. Выберем следующие параметры алгоритма: размер начальной популяции $Ip = 20$; максимальное количество популяций $K = 2000$; тип клонирования – «равномерное»; параметр клонирования $Nc = 10$; параметр мутации $\gamma = 100$; порог близости между популяциями $\varepsilon = 0.001$; порог близости между клетками $\sigma = 0.5$; процент добавляемых клеток $d = 40\%$.

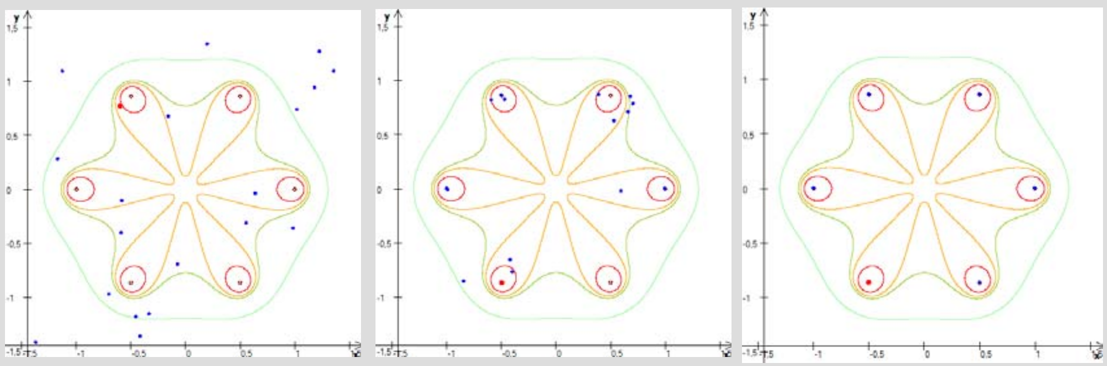
На рис. представлена начальная ($k = 0$), промежуточные $k = 123, k = 280$, ($k = 549$) и конечная ($k = 1345$) популяции клеток. Результаты работы расширенного метода ИИС: сформировано популяций $k = 1345$; количество локальных поисков $m = 27$; размер конечной популяции $Np = 435$; клетка с наилучшей приспособленностью $(x^*; y^*)^T = (0;0)^T$; наилучшая приспособленность $f(x^*; y^*) = 19,999943$; отклонение от точного решения $\Delta = 5.7 \cdot 10^{-5}$.



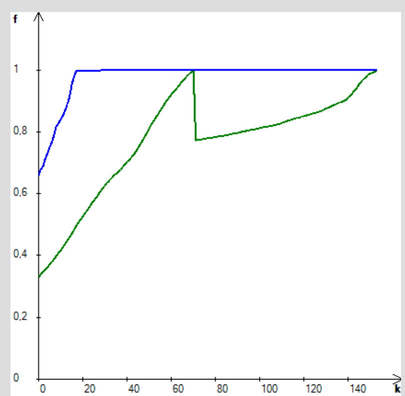
Начальная, промежуточные и конечная популяции в примере 2

Пример 3. Рассмотрим корневую функцию (табл. П.1). Зададим множество допустимых решений $x, y \in [-1,5;1,5]$. Выберем следующие параметры алгоритма: размер начальной популяции $Ip = 20$; максимальное количество популяций $K = 500$; тип клонирования – «равномерное»; параметр клонирования $Nc = 10$; параметр мутации $\gamma = 100$; порог близости между популяциями $\varepsilon = 0.00001$; порог близости между клетками $\sigma = 0.5$; процент добавляемых клеток $d = 40\%$.

На рис.представлена начальная ($k = 0$), промежуточная ($k = 50$) и конечная ($k = 152$) популяции клеток. Результаты работы расширенного метода ИИС: сформировано популяций $k = 152$; количество локальных поисков $m = 2$; размер конечной популяции $Np = 6$; клетка с лучшей приспособленностью $(x^*; y^*)^T = (-0,5; -0,866)^T$; лучшая приспособленность $f(x^*; y^*) = 0,9998$; отклонение от точного решения $\Delta = 0,0002$.



Начальная, промежуточная и конечная популяции в примере 2.3



Графики изменения средней (нижний)
и наилучшей (верхний) приспособленности популяции

I. B. МЕТОДЫ РАССЕИВАНИЯ

Постановка задачи

Требуется найти глобальный условный минимум функции $f(x)$ на множестве D, т.е. такую точку $x^* \in D$, что $f(x^*) = \min\limits_{x \in D} f(x)$, где $x = (x_1, x_2, ..., x_n)^T$, $D = \{x \mid x_i \in [a_i, b_i], i = 1, 2, ..., n\}$.

Стратегия поиска решения

В методе рассеивания рассматриваемая целевая функция $f(x)$ называется *функцией приспособленности*, а вектор параметров $x = (x_1, x_2, ..., x_n)^T$ целевой функции – *особью*. Чем меньше значение целевой функции $f(x)$, тем более особь x приспособлена, т.е. подходит в качестве решения.

Свое название метод берет от способа формирования начального набора возможных решений, называемого *базовым множеством особей*. Генерация базового множества построена таким образом, чтобы особи в нем были достаточно различны (рассеяны) между собой. Это позволяет эффективно исследовать все множество допустимых решений в поставленной задаче.

Процедура поиска начинается с *генерации базового множества особей* A. Для этого сначала на отрезке $[a_i, b_i]$ изменения каждой координаты выделяются s подынтервалов одинаковой длины. В базовое множество последовательно добавляется по одной особи:

1) для каждого отрезка $[a_i, b_i]$ генерируется номер j_i подынтервала с вероятностью, обратно пропорциональной числу раз, которое этот подынтервал уже выбирался;

2) генерируются значения координат x_i, $i = 1, ..., n$ случайным образом внутри выбранных подынтервалов;

3) если минимальное расстояние между сформированной особью и особями базового множества больше заданной величины σ, то особь добавляется в базовое множество A. Иначе – особь не добавляется, процесс рассеивания продолжается с п.1.

Процесс формирования базового множества гарантирует разнообразие входящих в него особей и продолжается до тех пор, пока не будет выбрано $Asize$ особей.

При решении задачи используются конечные наборы $I=\{x^j=(x_1^j,x_2^j,\ldots,x_n^j)^T, j=1,2,\ldots,Np\}\subset D$ возможных решений, называемые *популяциями*, где x^j – *особь* с номером j, Np – *размер популяции*.

Идеи рассеивания применяются также и при *формировании начальной популяции*: $Np=b_1+b_2$ особей выбираются из базового множества особей A, при этом –

b_1 особей выбираются по качеству (наилучшие особи по величине функции приспособленности),

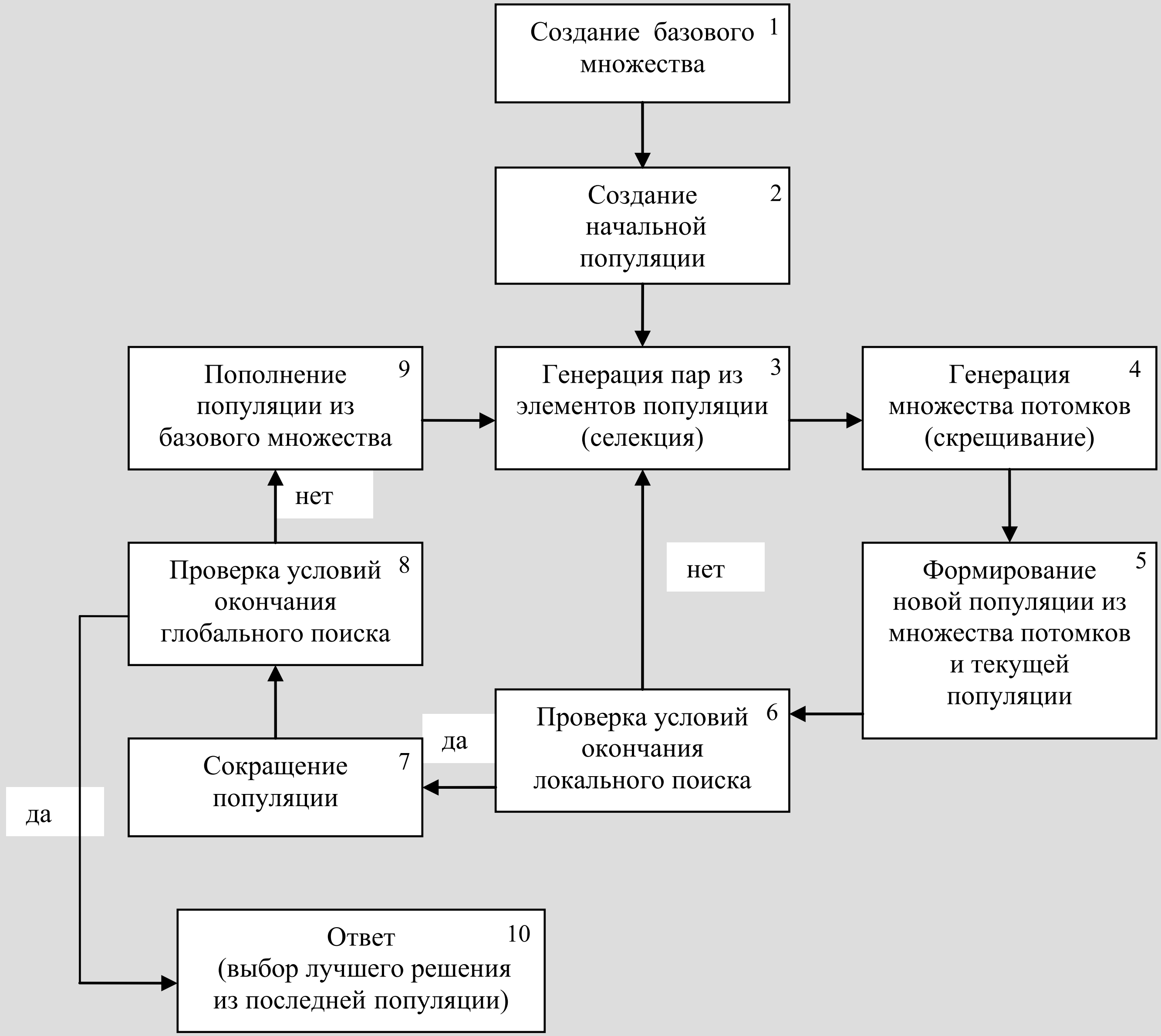
b_2 особей выбираются по расстоянию (суммарное расстояние от них до уже имеющихся в начальной популяции особей должно быть минимально).

Метод рассеивания имитирует эволюцию начальной популяции $I_0=\{x^j, j=1,2,\ldots,Np \mid x^j=(x_1^j,x_2^j,\ldots,x_n^j)^T\in D\}$ и представляет собой итерационный процесс, исследующий множество D. Во время работы метода на каждой итерации к популяции применяются биологические операторы: селекция и скрещивание, после чего происходит замена особей с низким уровнем приспособленности на новые. Таким образом, формируется новая популяция. Метод заканчивает работу после того, как будет исчерпано заданное количество вычислений значения функции приспособленности. В качестве приближенного решения задачи из последней популяции выбираются особи, которым соответствует наименьшее значение целевой функции. Следует отметить, что размер базового множества особей A должен быть достаточно большим, чтобы обеспечить работу метода до выполнения условий окончания.

Метод рассеивания сочетает в себе локальный и глобальный поиск решения.

Глобальный поиск реализуется в виде циклического итерационного процесса. Каждая итерация глобального поиска представляет собой локальный поиск. В конце каждой итерации происходит сокращение популяции: удаляются b_2 особей с наихудшей приспособленностью, после чего проверяются условия окончания работы метода. Если условия окончания не выполнены, то к популяции присоединяются b_2 новых особей из базового множества таким же образом, как и при формировании начальной популяции, после чего начинается новая итерация глобального поиска. Если условие окончания работы метода выполнено, то в качестве приближенного решения задачи из последней популяции выбираются особи, которым соответствует наименьшее значение целевой функции.

Локальный поиск реализуется в виде циклического итерационного процесса, во время которого к популяции применяются биологические операторы: селекция и скрещивание. Во время селекции из особей текущей популяции составляются всевозможные родительские пары, каждая из которых во время скрещивания порождает новое решение, называемое потомком. Все потомки помещаются в множество потомков, после чего происходит их сравнение с особями из текущей популяции и принимается решение о замене особей потомками. Таким образом, происходит смена популяции на новую, к которой, если условия окончания локального поиска не выполнены, опять применяются биологические операторы, и т.д. до выполнения условия окончания локального поиска. Средняя приспособленность популяции при этом будет расти. Если условие окончания локального поиска выполнено, то начинается новая итерация глобального поиска.



Общая схема работы метода рассеивания

Модификации.

1. *Мутация* применяется к множеству потомков во время его формирования. Среди потомков выделяются $Np = b_1 + b_2$ наилучших решений, для каждого из которых применяется процедура улучшения. Данная процедура может быть осуществлена несколькими способами, использующими идеи локального линейного поиска, табу-поиска, а также метода деформируемого многогранника с его модификациями. Если в результате мутации потомка находится лучшее решение, то оно его заменяет, иначе потомок остается неизменным.

2. В качестве процедуры локального поиска может применяться *процедура перекоммутации пути* (Path-relinking). При этом из популяции обычно выделяются три элемента. Далее делается несколько исследующих шагов в направлении от первого элемента ко второму и из полученных решений выбирается наилучшее. Далее такой же поиск производится из найденного решения в направлении третьего элемента.

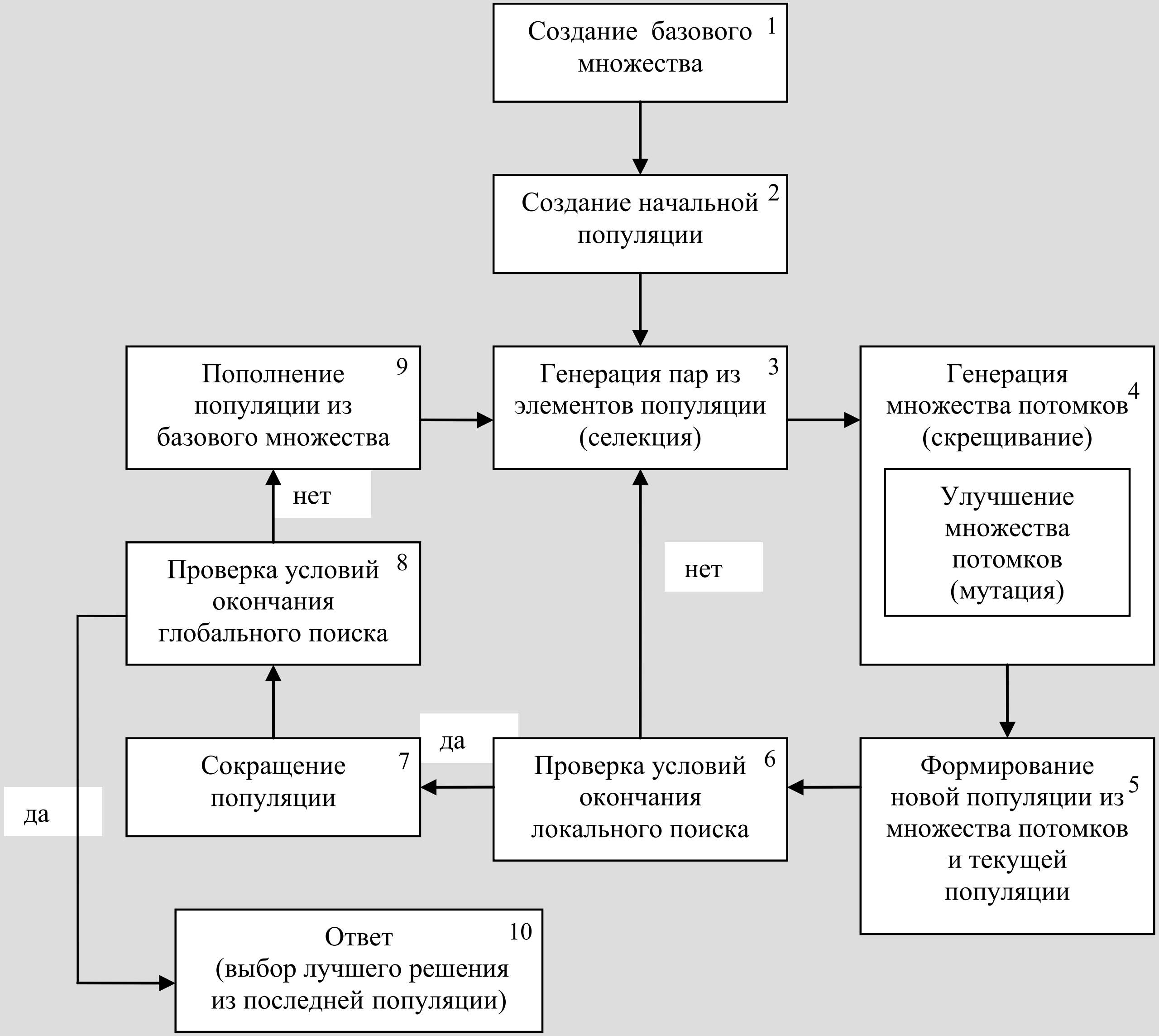


Схема работы модифицированного метода рассеивания

Алгоритм решения задачи

Шаг 1. *Создание базового множества* A.

Шаг 1.1. Задать максимальное количество особей в базовом множестве $Asize$, количество подынтервалов s, параметр рассеивания σ, максимальное количество вычислений целевой функции N. Задать $k=0$ (счетчик количества особей в базовом множестве) и $p=0$ (счетчик числа вычислений функции приспособленности).

Шаг 1.2. На каждом отрезке $[a_i,b_i]$, $i=1,\dots,n$, где n – число переменных функции приспособленности, выделить s подынтервалов. Для каждой координаты $i=1,\dots,n$ и для каждого номера подынтервала $j=1,\dots,s$ задать количество попаданий $freq(i,j)=0$.

Шаг 1.3. Для каждого подынтервала с номером j, $j=1,\dots,s$, на каждом отрезке $[a_i,b_i]$, $i=1,\dots,n$, вычислить вероятность его выбора:

$$P_j^i = \frac{\sum\limits_{k=1,k\neq j}^{s}[1+freq(i,k)]}{(s-1)(s+\sum\limits_{k=1}^{s} freq(i,k))}.$$

Шаг 1.4. Для каждой координаты $i=1,\dots,n$ с вероятностью P_j^i выбрать подынтервал с номером $m_i \in \{1,\dots,s\}$, и, используя равномерный закон распределения на выбранных подынтервалах, генерировать значения координат y_i и, как следствие, особь $y \in D$.

Шаг 1.5. Если $y \notin A$ и $d(y,A) > \sigma$, где $d(y,A)=\min\limits_{x\in A} d(x,y)$, $d(x,y)=\sqrt{\sum\limits_{i=1}^{n}(x_i-y_i)^2}$, то добавить особь y к множеству A: $A = A \cup \{y\}$, положить $k=k+1$; иначе перейти к шагу 1.4.

Шаг 1.6. Для каждой координаты $i=1,\dots,n$ изменить количество попаданий $freq(i,m_i)=freq(i,m_i)+1$.

Шаг 1.7. Если $k=Asize$, процесс генерации базового множества завершить, перейти к шагу 1.8; иначе перейти к шагу 1.3.

Шаг 1.8. Для всех особей множества A вычислить значение функции приспособленности $f(x^k)$, $k=1,\dots,Asize$, расположить особи по возрастанию соответствующих им значений функции приспособленности: $A=\{x^{(1)},\dots,x^{(Asize)}\}$, где $f(x^{(1)})=f_{\min}$. Положить $p=Asize$.

Результатом шага 1 является сформированное упорядоченное базовое множество A с особями, достаточно отличающимися друг от друга.

Шаг 2. *Создание начальной популяции.*

Шаг 2.1. Задать количество «качественных» особей в популяции b_1 и количество добавляемых по расстоянию особей b_2, $t = 0$ – счетчик популяций.

Шаг 2.2. Составить начальную популяцию I_0 из b_1 первых (лучших по значению функции приспособленности) особей базового множества A и удалить их из A.

Шаг 2.3. Данный шаг выполнить b_2 раз. Для каждой особи множества A вычислить расстояние до особей начальной популяции: $d(x^j, I_0) = \sum_{x \in I_0} d(x^j, x)$. Добавить в начальную популяцию особь с минимальным расстоянием до начальной популяции и удалить ее из базового множества A.

Шаг 2.4. Расположить особи начальной популяции по возрастанию соответствующего им значения функции приспособленности: $I_0 = \{x^{(1)}, ..., x^{(Np)}\}$, где $Np = b_1 + b_2$, $f(x^{(1)}) = f_{\min}$.

Результатом шага 2 является упорядоченная начальная популяция I_0.

Шаг 3. *Генерация родительских пар из элементов популяции (селекция).*

Сформировать множество Set из пар особей, входящих в текущую популяцию (число пар равно $\dfrac{Np \cdot (Np-1)}{2}$, где $Np = b_1 + b_2$); при очередном формировании в множество Set должна входить хотя бы одна новая пара.

Результатом шага 3 является множество Set, состоящее из родительских пар.

Шаг 4. *Генерация множества потомков (скрещивание).*

Создать пустое множество потомков $ChildSet$. Для каждой пары $\langle x; y \rangle$ из множества Set выполнить шаги 4.1–4.4:

Шаг 4.1. Осуществить линейный поиск вдоль направления, определяемого прямой, проходящей через точки x и y: $z_i(\lambda_i) = x + \lambda_i(y - x)$, $i = 1, 2, 3$, $\lambda_1 = \frac{1}{2}$, $\lambda_2 = \frac{4}{3}$, $\lambda_3 = -\frac{1}{3}$.

Шаг 4.2. Вычислить значения функции приспособленности $f(z_1)$, $f(z_2)$, $f(z_3)$. Выбрать потомка z^*, которому соответствует минимальное значение функции приспособленности. Положить $p = p + 3$.

Шаг 4.3. Добавить потомка z^* в множество потомков $ChildSet$. Удалить пару $\langle x; y \rangle$ из множества Set.

Шаг 4.4. При использовании модификаций применить один из методов улучшения множества потомков $ChildSet$, описанных далее. Пусть при использовании метода производилось Bp вычислений функции приспособленности, тогда положить $p = p + Bp$.

Результатом шага 4 является множество потомков $ChildSet$.

Шаг 5. *Формирование новой популяции.*

Шаг 5.1. Сформировать популяцию $I_{t+1} = I_t$. Положить $NewSolution = false$, $t = t + 1$. Для каждого потомка y из множества потомков $ChildSet$ выполнить:

Шаг 5.2. Для каждой особи x^i, $i = 1,..,Np$, из новой популяции I_t вычислить расстояние $d(x^i, y)$. Выбрать решение x^y, которому соответствует наименьшее расстояние $d(x^y, y) = \min\limits_{x \in I_t} d(x, y)$.

Шаг 5.3. Если выполнено одно из условий:

а) $f(y) < f(x^1)$;

б) $f(y) < f(x^{Np})$, где $Np = b_1 + b_2$, и $d(x^y, y) > \sigma$,

то заменить особь x^{Np} в популяции I_t на потомка y: $I_t = I_t \setminus \{x^{Np}\} \cup \{y\}$, положить $NewSolution = true$; иначе перейти к следующему потомку.

Результатом шага 5 является новая популяция.

Шаг 6. *Проверка условий окончания локального поиска.*

Если $NewSolution = true$, то локальный поиск продолжить, перейти к шагу 3.

Если $NewSolution = false$, то локальный поиск закончить, перейти к шагу 7.

Шаг 7. *Сокращение популяции.*

Расположить особи текущей популяции по возрастанию соответствующего им значения функции приспособленности: $I_t = \{x^{(1)},...,x^{(Np)}\}$, где $Np = b_1 + b_2$, $f(x^{(1)}) = f_{\min}$. Удалить b_2 последних особей (с наихудшим значением функции приспособленности).

Результатом шага 7 является сокращенная популяция.

Шаг 8. *Проверка условий окончания глобального поиска.*

Если $p < N$, то глобальный поиск продолжить, перейти к шагу 9;

Если $p \ge N$, то глобальный поиск завершить, перейти к шагу 10.

Шаг 9. *Пополнение популяции из базового множества.*

Шаг 9.1. Выполнить шаг b_2 раз. Для каждой особи множества A вычислить расстояние до особей текущей популяции: $d(x^j, I_t) = \sum\limits_{x \in I_t} d(x^j, x)$. Добавить в популяцию особь с минимальным расстоянием до популяции и удалить ее из базового множества A.

Шаг 9.2. Расположить особи популяции по возрастанию соответствующего им значения функции приспособленности: $I_t = \{x^{(1)},...,x^{(Np)}\}$, где $Np = b_1 + b_2$, $f(x^{(1)}) = f_{\min}$. Перейти к шагу 3.

Шаг 10. *Выбор решения из последней популяции.*

Закончить работу алгоритма. В качестве приближенного решения задачи $f(x^*) = \min\limits_{x \in D} f(x)$ выбрать особь с наименьшим значением функции приспособленности из текущей популяции: $x^* \cong \tilde{x}^* = \arg\min\limits_{j=1,...,Np} f(x^j)$.

Модификации

Локальный линейный поиск. Метод имеет параметр – шаг поиска h, который необходимо задать в начале работы метода рассеивания. Ниже приведен алгоритм локального линейного поиска.

1. Выделить среди множества потомков $ChildSet$ Np особей с наилучшей приспособленностью. Если в множестве меньше, чем Np потомков, выделить все множество.

2. Для каждого из выделенных потомков выполнить процедуру локального линейного поиска (шаги 2.1–2.4), если она не была выполнена до этого в процессе работы метода.

2.1. Упорядочить переменные x_i, $i = 1,..,n$, мутирующего потомка x случайным образом. Положить $New = false$ – индикатор наличия улучшений, $j = 0$ – номер переменной в упорядоченном списке.

2.2. Пусть на j-м месте в списке стоит i-я переменная. «Просканировать» множество, то есть вычислить значение функции приспособленности для особей из множества $ls(x,h,i) = \{y \in R^h \mid y = x + khe_i, k \in Z, a \le y \le b\}$, e_i – единичный орт, состоящий из всех нулей и одной единицы на i-м месте; $a = (a_1, a_2, ..., a_n)^T$, $b = (b_1, b_2, ..., b_n)^T$ – векторы соответственно нижних и верхних границ множества D.

2.3. Если в результате сканирования находится особь y, такая что $f(y) < f(x)$, то потомка x заменить особью y и положить $New = true$.

2.4. Если $j < n$, то положить $j = j + 1$ и перейти к шагу 2.2. Если $j = n$, то если $New = true$, положить $j = 0$, $New = false$, перейти к шагу 2.2, иначе процесс локального линейного поиска завершить.

Табу-поиск. В качестве улучшающего метода в методе рассеивания можно использовать табу-поиск. Дополнительно задаваемые параметры метода – количество улучшаемых направлений ts, максимальный размер табу-листа $tenure$, шаг поиска h. В начале работы метода рассеивания необходимо задать значения дополнительных параметров, а также создать пустой табу-лист. Далее, после каждой глобальной итерации табу-поиска, в табу-лист необходимо добавлять новую информацию.

1. Выделить среди множества потомков $ChildSet$ Np особей с наилучшей приспособленностью. Если в множестве меньше, чем Np потомков, выделить все множество.

2. Для каждого из выделенных потомков выполнить процедуру улучшения (шаги 2.1–2.5) (глобальная итерация табу-поиска).

2.1. Для каждого номера i переменной, которая не находится в табу-листе, найти величину привлекательности поиска: $A(x,i)=\max\limits_i[f(x)-f(x^{i+h}),f(x)-f(x^{i-h})]$, где $x^{i+h}=(x_1,..,x_i+h,...,x_n)$, $x^{i-h}=(x_1,..,x_i-h,...,x_n)$.

2.2. Упорядочить переменные по убыванию величины соответствующей привлекательности $A(x,i)$: первой переменной соответствует наибольшее значение $A(x,i)$. Положить $j=0$ – номер переменной в упорядоченном списке.

2.3. Пусть на первом месте в списке стоит i-я переменная. «Просканировать» множество $ls(x,h,i)$ как в локальном линейном поиске. Выбрать лучшую особь в данном множестве x' и заменить ею потомка x (даже если $f(x')>f(x)$). Занести i-ю переменную в табу-лист на $tenure$ глобальных итераций (после $tenure$ глобальных итераций i-ю переменную из табу-листа удалить). Положить $j=j+1$.

2.4. Пусть на j-м месте в списке стоит k-я переменная. «Просканировать» множество $ls(x',h,k)$ как в локальном линейном поиске. Выбрать лучшую особь в данном множестве x''. Если $f(x'')<f(x')$, то заменить x' на особь x''. Занести k-ю переменную в табу-лист на $tenure$ глобальных итераций.

2.5. Если $j<ts$, то положить $j=j+1$ и перейти к шагу 2.4. Если $j=ts$, то процесс завершить.

Во время работы табу-поиска размер табу-листа изменяется. До того как будет выполнено $tenure$ глобальных итераций, он равен количеству выполненных итераций, а после выполнения $tenure$ глобальных итераций становится постоянным и равным $tenure$.

Метод деформируемого многогранника и его модификация. Дополнительно задаваемые параметры в этом случае – шаг поиска h и параметры метода деформируемого многогранника.

1. Выделить среди множества потомков $ChildSet$ Np особей с наилучшей приспособленностью. Если в множестве меньше, чем Np потомков, выделить все множество.

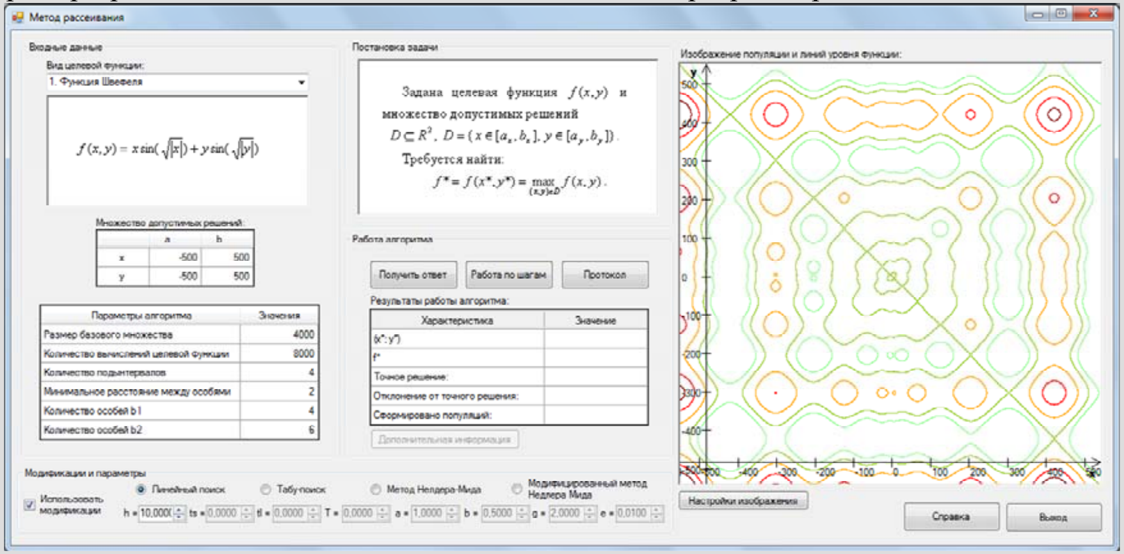
2. Для каждого из выделенных потомков выполнить шаги 2.1–2.2.

2.1. Для текущей особи x находится n пробных точек y^i, $i=1,..,n$: $y^i = x + hd_i$, где $d_i = e_i$ или $d_i = -e_i$ (знак выбирается случайным образом), e_i – единичный орт, состоящий из всех нулей и одной единицы на i-м месте. Формируется выпуклый многогранник $S = \{x, y^1, ..., y^n\}$.

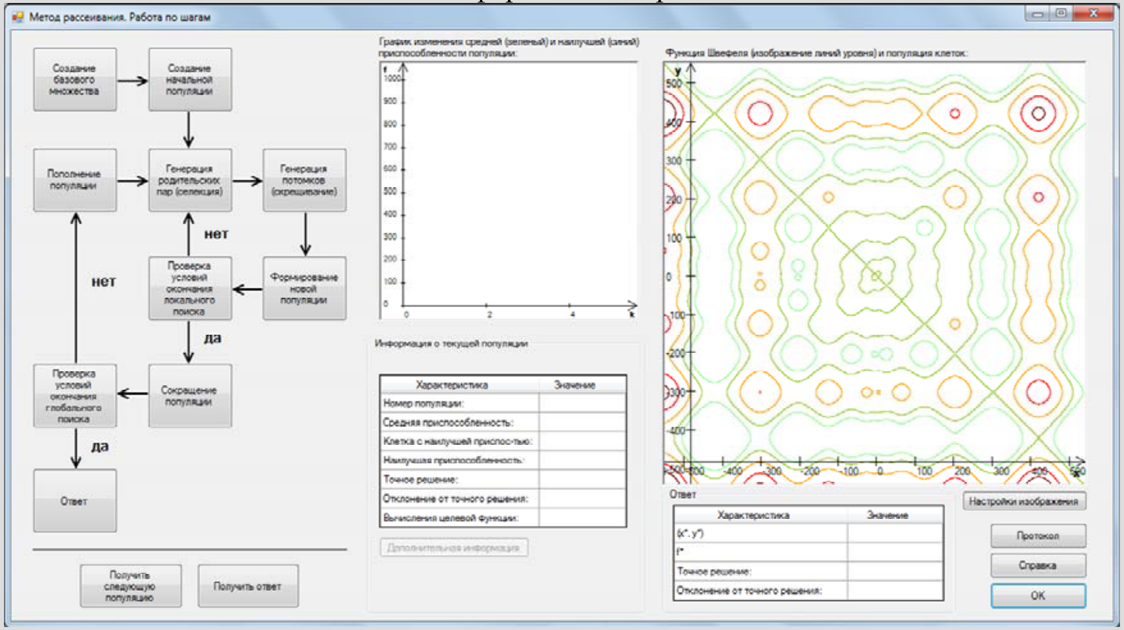
2.2. Выполнить метод деформируемого многогранника с проверкой принадлежности пробных точек множеству допустимых решений D.

Возможна модификация данного метода с применением табу-областей. Во время работы метода используется табу-лист заданной длины $NumSol$, в который заносятся последние $NumSol$ особей, к которым был применен метод. Вокруг особей, входящих в табу-лист, образуются табу-области, представляющие собой в двумерном случае круги заданного радиуса T. При этом в начале работы метода среди множества потомков $ChildSet$ выбираются особи, которые не лежат ни в одной табу-области.

Среда разработки Microsoft Visual Studio 2010, язык программирования C# .

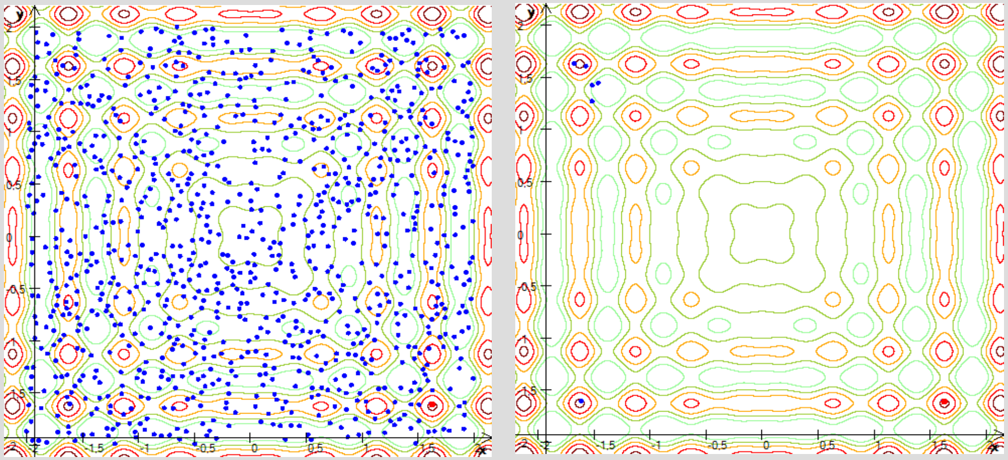


Главная форма метода рассеивания

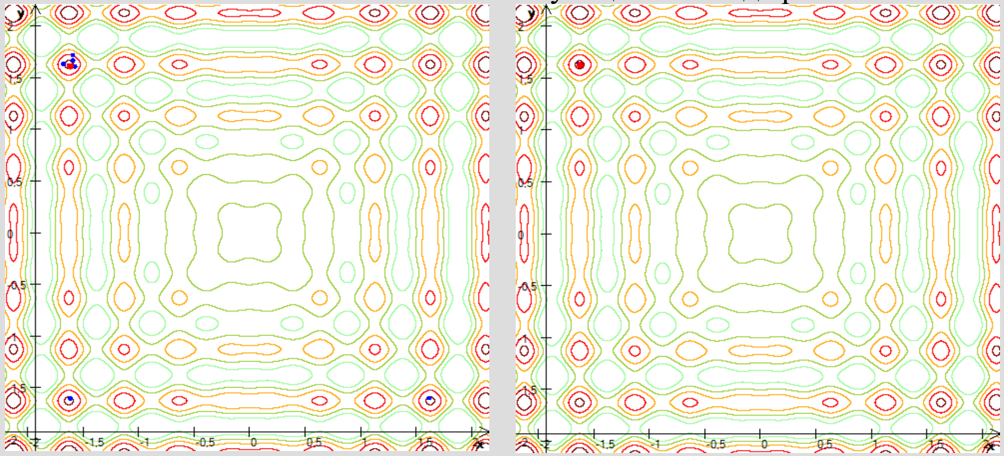


Форма работы метода рассеивания по шагам

Пример. Рассмотрим мульти-функцию. Зададим множество допустимых решений $x, y \in [-2;2]$. Выберем следующие параметры алгоритма: размер базового множества $Asize = 1000$; максимальное количество вычислений целевой функции $N = 5000$; количество подынтервалов $s = 4$; минимальное расстояние между особями $\sigma = 0.05$; количество «качественных» особей $b_1 = 4$; количество особей, добавляемых «по расстоянию» $b_2 = 4$; модификации не используются.



Базовое множество и начальная популяция в методе рассеивания



Промежуточная и конечная популяции в методе рассеивания

Результаты работы метода рассеивания: сформировано популяций $k = 63$; клетка с наилучшей приспособленностью $(x^*; y^*)^T = (-1,6283;1,6269)^T$; наилучшая приспособленность $f(x^*; y^*) = 4,253355$; отклонение от точного решения $\Delta = 5,3 \cdot 10^{-4}$.

I. Г. ЭВОЛЮЦИОННАЯ СТРАТЕГИЯ ПРЕОБРАЗОВАНИЯ КОВАРИАЦИОННОЙ МАТРИЦЫ

Постановка задачи

Требуется найти глобальный условный минимум целевой функции $f(x)$ на множестве D, т.е. такую точку $x^* \in D$, что $f(x^*) = \min\limits_{x \in D} f(x)$, где $x = (x_1, x_2, ..., x_n)^T$, $D = \{x \mid x_i \in [a_i, b_i], i = 1, 2, ..., n\}$.

Стратегия поиска решения

Подобно другим эволюционным методам, в эволюционной стратегии на каждой итерации образуется новая популяция. Генерация новых особей, так называемая мутация, происходит случайным образом согласно формируемому эволюционной стратегией распределению. Лучшие найденные решения участвуют в дальнейшем эволюционном процессе.

Для генерации новых особей (векторов) в эволюционных стратегиях применяется нормальное распределение, параметры которого влияют на эффективность работы реализующего стратегию алгоритма. Особенностью эволюционной стратегии является изменение параметров распределения на каждой итерации в соответствии с успешностью поиска. Стратегия сформирована таким образом, чтобы преобразовывать текущие параметры распределения таким образом, чтобы они как можно лучше подходили для успешной работы на последующих шагах.

В эволюционных стратегиях преобразования ковариационной матрицы рассматривается целевая функция $f(x)$, называемая, подобно всем эволюционным методам, *функцией приспособленности*. Вектор параметров $x = (x_1, x_2, ..., x_n)^T$ целевой функции называется *особью*. Каждый вектор $x = (x_1, x_2, ..., x_n)^T \in D$ является возможным решением поставленной оптимизационной задачи.

При решении задачи глобальной оптимизации используются конечные наборы $I = \{x^j = (x_1^j, x_2^j, \ldots, x_n^j)^T, j = 1, 2, \ldots, \lambda\} \subset D$ возможных решений, называемые *популяциями*, где x^j – особь с номером j, λ – *размер популяции*. Применение эволюционной стратегии сводится к исследованию множества D при помощи перехода от одной популяции к другой. Чем меньше значение целевой функции $f(x^j)$, тем более особь x^j приспособлена.

Рассматриваемая стратегия имитирует эволюцию начальной популяции $I_0 = \{x^j, j = 1, 2, \ldots, \lambda \mid x^j = (x_1^j, x_2^j, \ldots, x_n^j)^T \in D\}$, и представляет собой итерационный процесс. На каждой итерации генерируется новая популяция согласно сформированному стратегией распределению. После этого производится анализ новой популяции и корректировка параметров распределения: шага σ и ковариационной матрицы C, после чего осуществляется переход на новую итерацию. Процедура поиска завершается после того, как сформируется заданное количество популяций. В качестве приближенного решения задачи из последней популяции выбирается особь, которой соответствует наименьшее значение функции приспособленности.

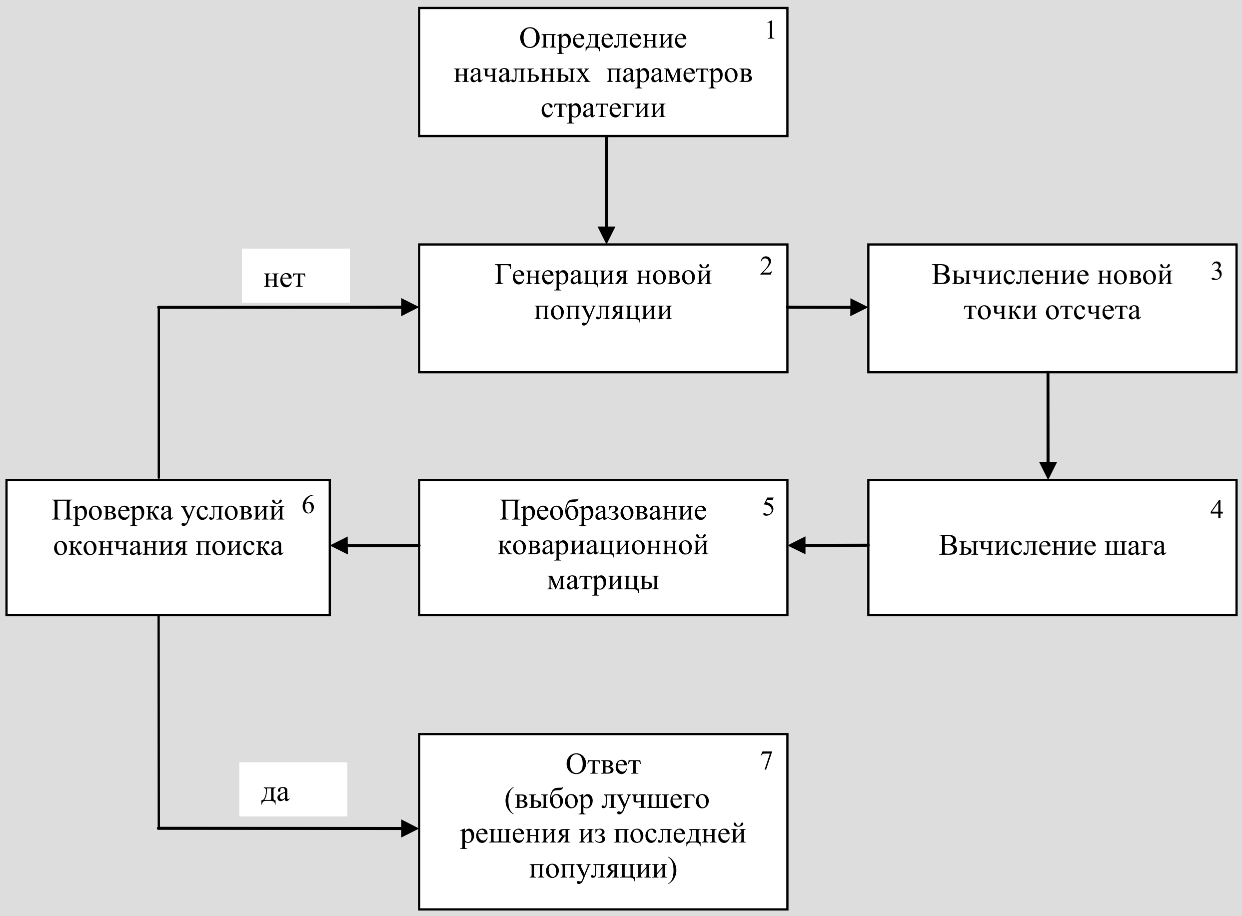


Схема реализации эволюционной стратегии преобразования ковариационной матрицы

Модификация.

В процессе реализации модифицированной эволюционной стратегии после создания начальной популяции на всех итерациях для каждой особи согласно сформированному распределению генерируется несколько потомков. Если значение функции приспособленности у потомка лучше, чем у его родительской особи, то в новую популяцию переходит потомок, а иначе – родительская особь. Стратегия включает в себя анализ успешности потомков и корректировку параметров распределения для каждой особи: шага σ_i и ковариационной матрицы C_i, $i = 1,...,\lambda$. После этого осуществляется переход на новую итерацию.

Схема реализации модифицированной эволюционной стратегии преобразования ковариационной матрицы

Алгоритм решения задачи

Шаг 1. Задать начальные значения параметров:

– сопряженный показатель роста $p_\sigma^{(0)} = 0$;

– показатель роста $p_c^{(0)} = 0$;

– ковариационную матрицу $C^{(0)} = I$;

– величину шага $\sigma^{(0)} = \frac{1}{2} \min\limits_{i=1,\ldots,n} (b_i - a_i)$;

– количество особей в популяции λ;

– количество элитных особей $\mu = \left[\frac{\lambda}{2} \right]$;

– количество популяций K;

– веса $\omega_i = \dfrac{\ln(\mu+1) - \ln(i)}{\sum\limits_{j=1}^{\mu} [\ln(\mu+1) - \ln(j)]}, i = 1,\ldots,\mu$ или $\omega_i = \dfrac{1}{\mu}, i = 1,\ldots,\mu$;

– весовой коэффициент $\mu_{eff} = \mu_{\text{cov}} = \dfrac{1}{\sum\limits_{i=1}^{\mu} \omega_i^2}$;

– параметры памяти $c_\sigma = \dfrac{\mu_{eff} + 2}{n + \mu_{eff} + 3}$, $c_c = \dfrac{4}{n+4}$;

– параметр затухания $d_\sigma = 1 + 2 \max \left\{ 0, \sqrt{\dfrac{\mu_{eff} - 1}{n+1}} - 1 \right\} + c_\sigma$;

– показатель скорости адаптации ковариационной матрицы

$$c_{\text{cov}} = \frac{2}{\mu_{\text{cov}} (n + \sqrt{2})^2} + \left(1 + \frac{1}{\mu_{\text{cov}}} \right) \min \left\{ 1, \frac{2\mu_{eff} - 1}{(n+2)^2 + \mu_{eff}} \right\}.$$

Случайным образом, используя равномерное распределение на множестве допустимых решений, сгенерировать начальную точку отсчета $\langle x \rangle_{\mathrm{w}}^{(0)}$. Задать $g = 1$ (счетчик популяций).

Шаг 2. Сгенерировать популяцию

$$x_k^{(g)} \sim \boldsymbol{N}\left(\langle x\rangle_{\mathrm{w}}^{(g-1)}, \left(\sigma^{(g-1)}\right)^2 C^{(g-1)}\right), \quad k = 1,\ldots,\lambda\,,$$

где $\boldsymbol{N}(m,D)$ – нормальное распределение случайного вектора с математическим ожиданием m и ковариационной матрицей D.

Упорядочить особи в популяции по возрастанию соответствующего им значения функции приспособленности (целевой функции). На первом месте должна стоять особь с наименьшим значением целевой функции.

Шаг 3. Вычислить новую точку отсчета – взвешенное среднее, определяемое по μ лучшим особям в популяции: $\langle x\rangle_{\mathrm{w}}^{(g)} = \sum_{i=1}^{\mu} \omega_i x_{i:\lambda}^{(g)}$, где $x_{i:\lambda}^{(g)}$ – *i*-я особь в упорядоченной популяции (*i*-я лучшая особь).

Шаг 4. Изменить величину шага. Для этого выполнить шаги 4.1 – 4.2.

Шаг 4.1. Вычислить новый сопряженный показатель роста

$$p_\sigma^{(g)} = (1-c_\sigma)p_\sigma^{(g-1)} + \sqrt{c_\sigma(2-c_\sigma)}B^{(g-1)}(D^{(g-1)})^{-1}(B^{(g-1)})^T \frac{\sqrt{\mu_{eff}}}{\sigma^{(g-1)}}\left(\langle x\rangle_{\mathrm{w}}^{(g)} - \langle x\rangle_{\mathrm{w}}^{(g-1)}\right)\,,$$

где ортогональная матрица $B^{(g-1)}$ и диагональная матрица $D^{(g-1)}$ находятся при помощи разложения ковариационной матрицы $C^{(g-1)}$: $C^{(g-1)} = B^{(g-1)}(D^{(g-1)})^2(B^{(g-1)})^T$, для которого можно использовать, например, метод вращений;

Шаг 4.2. Вычислить новую величину шага

$$\sigma^{(g)} = \sigma^{(g-1)} \cdot \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\left\|p_\sigma^{(g)}\right\|}{E\left(\left\|N(0,\boldsymbol{I})\right\|\right)} - 1\right)\right),$$

где $E\left(\left\|N(0,\boldsymbol{I})\right\|\right) = \sqrt{2}\Gamma(\tfrac{n+1}{2}) / \Gamma(\tfrac{n}{2}) \approx \sqrt{n}\left(1 - \frac{1}{4n} + \frac{1}{21n^2}\right)$ – ожидаемая длина вектора p_σ при условии случайного распределения его элементов, $E(\cdot)$ – знак математического ожидания, $\Gamma(\cdot)$ – гамма-функция, $\|\cdot\|$ – знак нормы вектора, $N(0,\boldsymbol{I})$ – нормальное распределение случайного вектора с нулевым математическим ожиданием и единичной ковариационной матрицей.

Шаг 5. Изменить ковариационную матрицу. Для этого выполнить шаги 5.1 – 5.2.

Шаг 5.1. Вычислить новый показатель роста

$$p_c^{(g)} = (1-c_c)p_c^{(g-1)} + H_\sigma^{(g)}\sqrt{c_c(2-c_c)}\,\frac{\sqrt{\mu_{eff}}}{\sigma^{(g-1)}}\left(\left\langle x\right\rangle_{\mathrm{w}}^{(g)} - \left\langle x\right\rangle_{\mathrm{w}}^{(g-1)}\right),$$

где $H_\sigma^{(g)} = 1$, если $\dfrac{\left\|p_\sigma^{(g)}\right\|}{\sqrt{1-(1-c_\sigma)^{2g}}} < \left(\dfrac{3}{2} + \dfrac{1}{n-0.5}\right)E\left(\left\|N(0,\boldsymbol{I})\right\|\right)$, и $H_\sigma^{(g)} = 0$ в противном случае.

Шаг 5.2. Изменить ковариационную матрицу

$$C^{(g)} = (1-c_{\mathrm{cov}})C^{(g-1)} + c_{\mathrm{cov}}\frac{1}{\mu_{\mathrm{cov}}}p_c^{(g)}\left(p_c^{(g)}\right)^T +$$

$$+c_{\mathrm{cov}}\left(1-\frac{1}{\mu_{\mathrm{cov}}}\right)\sum_{i=1}^{\mu}\frac{\omega_i}{(\sigma^{(g-1)})^2}\left(x_{i:\lambda}^{(g)} - x_{\mathrm{w}}^{(g-1)}\right)\left(x_{i:\lambda}^{(g)} - x_{\mathrm{w}}^{(g-1)}\right)^T.$$

Шаг 6. Положить $g = g+1$. Если $g < K$, то перейти к шагу 2, иначе перейти к шагу 7.

Шаг 7. Закончить работу алгоритма. В качестве приближенного решения задачи $f(x^*) = \min\limits_{x\in D} f(x)$ выбрать особь с наименьшим значением функции приспособленности из текущей популяции: $x^* \cong \tilde{x}^* = \arg\min\limits_{j=1,\ldots,\lambda} f(x_j)$.

Модифицированный алгоритм

Шаг 1. Задать параметры алгоритма:

– количество генерируемых каждой особью потомков s;

– количество особей в популяции λ;

– количество популяций K;

– заданный показатель роста $p_{succ}^{target} = \dfrac{1}{5+\sqrt{s}/2}$;

– предельный показатель роста $p_{thresh} = 0,44$;

– параметры памяти $c_p = \dfrac{p_{succ}^{target} s}{2 + p_{succ}^{target} s}$, $c_c = \dfrac{2}{n+2}$;

– параметр затухания $d = 1 + \dfrac{n}{2s}$;

– показатель скорости преобразования ковариационной матрицы

$$c_{\text{cov}} = \frac{2}{n^2+6}.$$

Шаг 2. Сгенерировать начальную популяцию $I_0 = \{a_1^{(0)},...,a_\lambda^{(0)}\}$, в которой каждая особь представляется набором $a_i^{(0)} = \{x_i^{(0)}, \overline{p}_{succ,i}^{(0)}, \sigma_i^{(0)}, p_{c,i}^{(0)}, C_i^{(0)}\}$, $i = 1,...,\lambda$, где

– $x_i^{(0)} \in R^n$ – хромосома, возможное решение задачи, которая генерируется случайным образом при помощи равномерного распределения на множестве допустимых решений;

– $\overline{p}_{succ,i}^{(0)} = p_{succ}^{target}$ – средний показатель роста;

– $\sigma_i^{(0)} = \dfrac{1}{2}\min\limits_{i=1,...,n}(b_i - a_i)$ – величина шага;

– $p_{c,i}^{(0)} = 0$ – показатель роста;

– $C_i^{(0)} = I \in R^{n\times n}$ – ковариационная матрица;

Задать $g = 0$ (счетчик итераций).

Шаг 3. Для каждой особи $a_i^{(g)}$, $i = 1,...,\lambda$ в популяции I_g:

1) сгенерировать s хромосом–потомков $x_{i,k}^{(g)} \sim \boldsymbol{N}\left(x_i^{(g)}, \left(\sigma^{(g)}\right)^2 C^{(g)}\right), k = 1,...,s$;

2) упорядочить потомков по возрастанию соответствующих им значений функции приспособленности;

3) вычислить долю успешных потомков: $p_{succ,i} = \dfrac{s_{succ,i}^{(g)}}{s}$, где $s_{succ}^{(g)}$ – количество потомков, для которых $f(x_{i,k}^{(g)}) \le f(x_i^{(g)})$, $k = 1,...,s$.

Шаг 4. Сгенерировать новую популяцию I_{g+1}. Для каждой особи $a_i^{(g+1)}$, $i=1,...,\lambda$ в новой популяции вычислить средний показатель роста и размер шага:

$$\overline{p}_{succ,i}{}^{(g+1)} = (1-c_p)\overline{p}_{succ,i}{}^{(g)} + c_p p_{succ,i}\,,$$

$$\sigma_i{}^{(g+1)} = \sigma_i{}^{(g)} \cdot \exp\left(\frac{1}{d}\cdot\frac{\overline{p}_{succ,i}{}^{(g+1)} - p_{succ}^{target}}{1-p_{succ}^{target}}\right).$$

Шаг 5. Для каждой особи $a_i^{(g)}$, $i=1,...,\lambda$ выполнить проверку.

Если первый потомок в упорядоченном множестве потомков лучше родительской хромосомы, т. е. $f(x_{i,1}^{(g)}) \le f(x_i^{(g)})$, то положить $x_i^{(g+1)} = x_{i,1}^{(g)}$, выполнить шаг 6.

Если $f(x_{i,1}^{(g)}) > f(x_i^{(g)})$, то положить $x_i^{(g+1)} = x_i^{(g)}$, $p_{c,i}^{(g+1)} = p_{c,i}^{(g)}$, $C_i{}^{(g+1)} = C_i{}^{(g)}$, перейти к шагу 7.

Шаг 6. Для каждой особи $a_i^{(g+1)}$, $i=1,...,\lambda$ в новой популяции изменить ковариационную матрицу.

Если $\overline{p}_{succ,i}{}^{(g+1)} < p_{thresh}$, тогда вычислить новый показатель роста и ковариационную матрицу:

$$p_{c,i}^{(g+1)} = (1-c_c)p_{c,i}^{(g)} + \frac{\sqrt{c_c(2-c_c)}}{\sigma_i{}^{(g)}}(x_i^{(g+1)} - x_i^{(g)})\,,$$

$$C_i{}^{(g+1)} = (1-c_{\mathrm{cov}})\, C_i{}^{(g)} + c_{\mathrm{cov}} p_{c,i}^{(g+1)}\left(p_{c,i}^{(g+1)}\right)^T.$$

Иначе, если $\overline{p}_{succ,i}{}^{(g+1)} \ge p_{thresh}$, то уменьшить показатель роста и вычислить новую ковариационную матрицу

$$p_c^{(g+1)} = (1-c_c)p_c^{(g)}\,,$$

$$C_i{}^{(g+1)} = (1-c_{\mathrm{cov}})C_i{}^{(g)} + c_{\mathrm{cov}}\left(p_{c,i}^{(g+1)}\left(p_{c,i}^{(g+1)}\right)^T + c_c(2-c_c)C_i{}^{(g)}\right).$$

Шаг 7. Положить $g = g+1$. Проверить условия окончания.

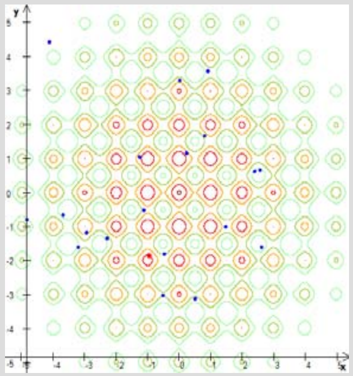
Если $g < K$, то перейти к шагу 3, иначе поиск прекратить, перейти к шагу 8.

Шаг 8. Закончить работу алгоритма. В качестве приближенного решения задачи $f(x^*) = \min\limits_{x\in D} f(x)$ выбрать особь с наименьшим значением функции приспособленности из текущей популяции: $x^* \cong \tilde{x}^* = \arg\min\limits_{j=1,...,\lambda} f(x_j)$.

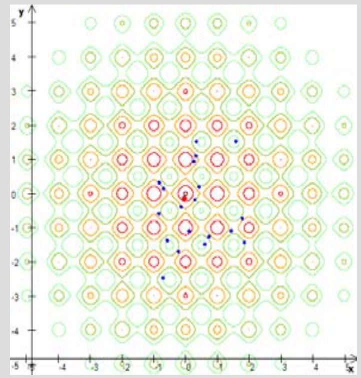
Пример 1. Рассмотрим функцию Растригина (табл. П.1). Зададим множество допустимых решений $x, y \in [-5;5]$. Выберем следующие параметры стратегии ***без модификаций***: количество популяций $K = 30$; количество особей в популяции $\lambda = 20$; параметр метода вращения $\varepsilon = 0,001$.

На рис. представлены начальная, промежуточные и конечная популяции особей.

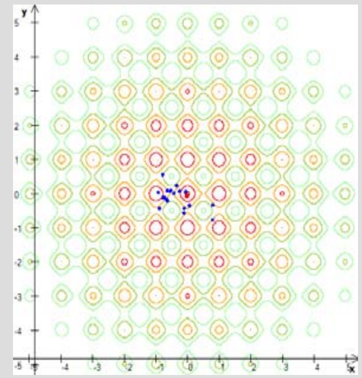
Результаты работы эволюционной стратегии: клетка с наилучшей приспособленностью $(x^*; y^*)^T = (0;0)^T$; наилучшая приспособленность $f(x^*; y^*) = 20$; отклонение от точного решения $\Delta = 0$.



$g = 0$

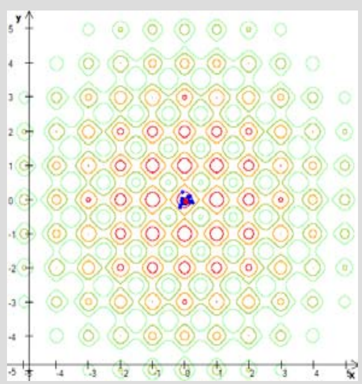


$g = 3$

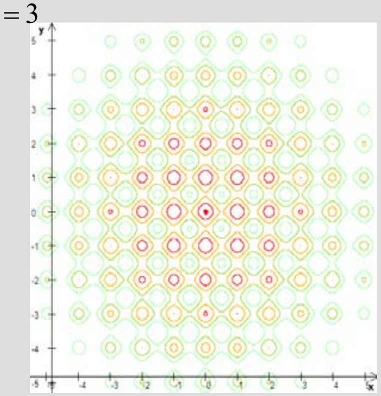


$g = 5$

$g = 3$



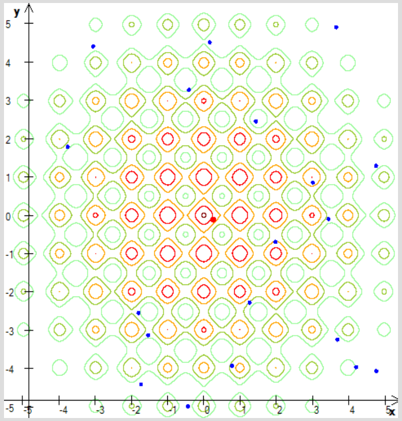
$g = 8$



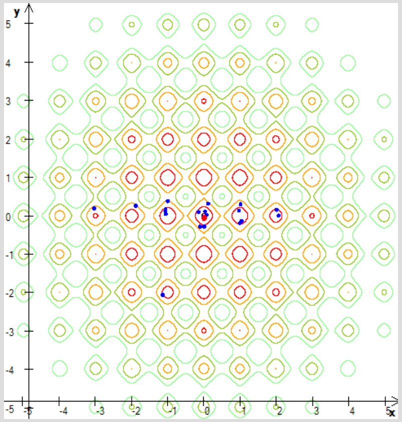
$g = 30$

Начальная, промежуточные и конечная популяции в примере 1

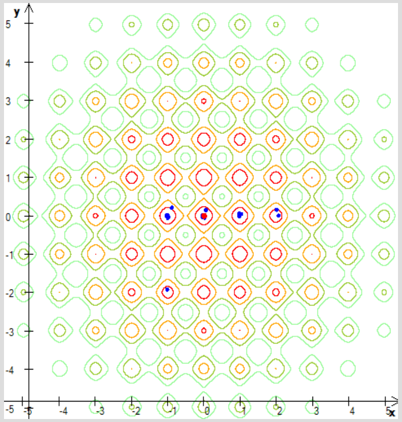
Пример 2. Решим задачу при помощи ***модификации*** эволюционной стратегии преобразования ковариационной матрицы при следующих параметрах: количество генерируемых каждой особью потомков $s = 5$; количество особей в популяции $\lambda = 20$; количество популяций $K = 30$.



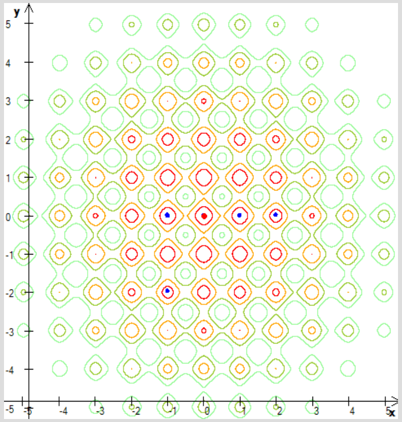
$g = 0$



$g = 10$



$g = 20$



$g = 30$

Начальная, промежуточные и конечная популяции в примере 2

Результаты работы модифицированной стратегии: клетка с наилучшей приспособленностью $(x^*; y^*)^T = (0;0)^T$; наилучшая приспособленность $f(x^*; y^*) = 20$; отклонение от точного решения $\Delta = 0$.

I. Г. МЕТОД ДИНАМИЧЕСКИХ СЕТОК

Постановка задачи

Требуется найти глобальный условный максимум функции $f(x)$ на множестве D, т.е. такую точку $x^* \in D$, что $f(x^*) = \max_{x \in D} f(x)$, где $x = (x_1, x_2, ..., x_n)^T$, $D = \{x \mid x_i \in [a_i, b_i], i = 1, 2, ..., n\}$.

Стратегия поиска решения

При работе метода происходит эволюция начальной популяции – смена одного поколения другим путем расширения и последующего сокращения популяции.

В методе динамических сеток популяция представляется в виде некоторой *сетки*, состоящей из набора решений, называемых *узлами*. В процессе поиска сетка подвергается изменениям: *расширению* – добавлению новых узлов в сетку, и *сокращению* – удалению узлов, расположенных слишком близко друг к другу.

В методе динамических сеток рассматривается целевая функция $f(x)$. Каждому узлу ставится в соответствие вектор параметров $x = (x_1, x_2, ..., x_n)^T$ целевой функции. Каждый вектор $x = (x_1, x_2, ..., x_n)^T \in D$ является возможным решением поставленной оптимизационной задачи. При решении задачи используются конечные наборы $I = \{x^j = (x_1^j, x_2^j, \ldots, x_n^j)^T, j = 1, 2, \ldots, P\} \subset D$ возможных решений, называемые *популяциями* или *сеткой*, где x^j – узел с номером j, P – количество узлов в сетке. Применение метода динамических сеток сводится к исследованию множества D при помощи изменения сетки и перехода от одной популяции к другой. Чем больше значение целевой функции $f(x^j)$, тем более узел x^j подходит в качестве решения.

Метод динамических сеток имитирует эволюцию начальной популяции $I_0 = \{x^j, j = 1, 2, \ldots, P \mid x^j = (x_1^j, x_2^j, \ldots, x_n^j)^T \in D\}$ и представляет собой итерационный процесс. Во время работы метода на каждой итерации происходит расширение (локальное, глобальное и дополнительное) и последующее сокращение сетки. Таким образом, формируется новая сетка. Критерием окончания поиска является достижение заранее заданного количества C вычислений целевой функции. В качестве приближенного решения задачи из последней популяции выбирается узел, которому соответствует наибольшее значение целевой функции.

В процессе расширения происходит добавление новых узлов в сетку. Стадия расширения состоит из нескольких этапов:

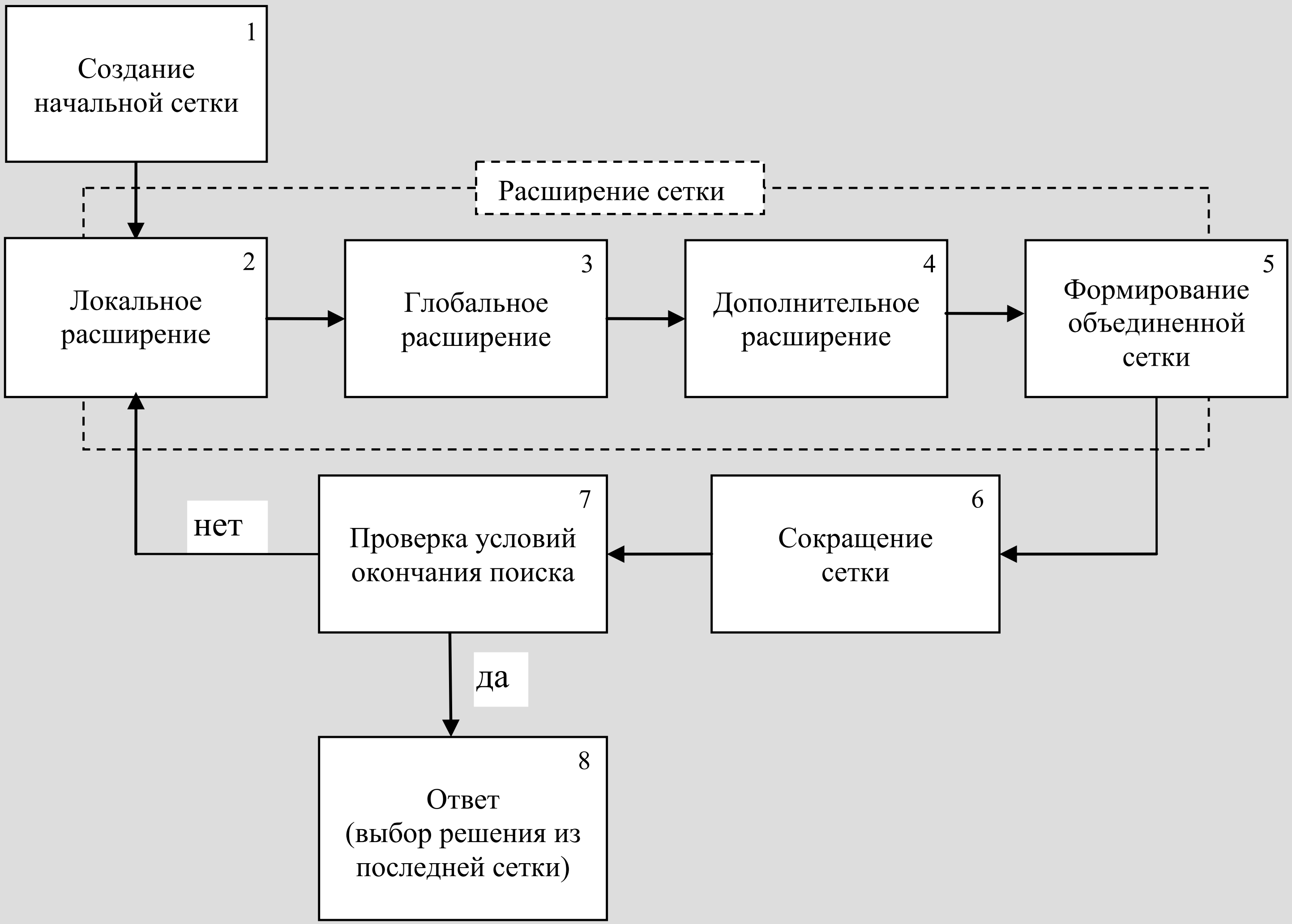
- локальное расширение,
- глобальное расширение,
- дополнительное расширение.

На этапе *локального расширения* в окрестности каждого p-го узла сетки, $p = 1, 2, ..., P$, выбирается некоторое заранее заданное число K ближайших по расстоянию узлов, называемых соседними узлами. Среди соседних узлов выбирается наилучший и, если данный соседний узел лучше p-го узла, то производится генерация нового узла в направлении наилучшего из K соседних узлов.

На этапе *глобального расширения* для всех узлов сетки, кроме наилучшего узла сетки x^{best}, производится генерация нового узла в направлении наилучшего узла сетки.

Если при локальном и глобальном расширении сгенерировано узлов меньше, чем заранее заданное число T, то выполняется *дополнительное расширение* сетки. Путем генерации новых узлов производится исследование новых участков множества допустимых решений.

На последующей за расширением *стадии сокращения* в методе динамических сеток происходит удаление слишком близких друг к другу решений, таким образом, стратегия метода направлена на поддержание достаточного разнообразия узлов в сетке.



Общая схема работы метода динамических сеток

АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ

Шаг 1. *Задать параметры метода*:

P – размер популяции – количество узлов в сетке на каждой итерации;

T – число новых узлов, генерируемых в процессе расширения;

K – количество соседних узлов каждого узла сетки;

C – максимальное число вычисление целевой функции.

Вычислить:

– вектор амплитуд множества допустимых решений по каждой координате $\Delta = (\Delta_1, \Delta_2, .., \Delta_n)^T$, где $\Delta_i = |b_i - a_i|$, $i = 1, ..., n$;

– центр множества допустимых решений – узел $x^m = (x_1^m, x_2^m, ..., x_n^m)^T$, где $x_i^m = \dfrac{a_i + b_i}{2}$, $i = 1, ..., n$.

Шаг 2. *Генерация начальной сетки*. С помощью равномерного распределения на множестве допустимых решений D сгенерировать сетку (начальную популяцию): $I_0 = \{x^p = (x_1^p, x_2^p, ..., x_n^p)^T,\ p = 1, 2, ..., P\} \subset D$.

Задать счетчик количества вычислений целевой функции $c = 0$, и счетчик популяций $t = 0$.

Шаг 3. *Вычислить* значения целевой функции во всех узлах сетки: $f(x^1), f(x^2), ..., f(x^P)$, и найти среди них наилучшее решение x^{best}, которому соответствует наибольшее значение целевой функции: $f(x^{best}) = \max\limits_{p=1,..,P} f(x^p)$. Положить $c = c + P$.

Шаг 4. *Локальное расширение сетки*. Генерация новых узлов по направлению к локальным экстремумам в окрестности каждого узла. Для каждого узла x^p, $p=1,...,P$ выполнить шаги 4.1.–4.4.

Шаг 4.1. Вычислить расстояния от p-го узла до всех остальных узлов сетки $d(x^p,x^j)=\sqrt{\sum_{i=1}^{n}\left(x_i^p-x_i^j\right)^2}$, $j=1,2,...,P$, $j\neq p$.

Шаг 4.2. Выбрать K узлов $\tilde{x}^1,\tilde{x}^2,...,\tilde{x}^K$, для которых расстояние d_{pj} до узла x^p минимально, данные узлы являются соседними узлами с узлом x^p.

Шаг 4.3. Среди соседних узлов найти наилучший узел $\tilde{x}^{best}$, которому соответствует наибольшее значение целевой функции $f(\tilde{x}^{best})=\max\limits_{k=1,...,K} f(\tilde{x}^k)$.

Шаг 4.4. Сравнить значения целевой функции в узлах $\tilde{x}^{best}$ и x^p: если $f(\tilde{x}^{best})<f(x^p)$, новый узел не генерировать; если $f(\tilde{x}^{best})\geq f(x^p)$, то сгенерировать новый узел $x^{loc}=(x_1^{loc},x_2^{loc},...,x_n^{loc})^T$:

– вычислить фактор близости текущего узла x^p и наилучшего соседнего с ним узла $\tilde{x}^{best}$: $Pr(x^p,\tilde{x}^{best})=\dfrac{1}{1+\left|f(x^p)-f(\tilde{x}^{best})\right|}$ (значение $Pr(x^p,\tilde{x}^{best})$ принадлежит отрезку $[0;1]$, чем больше значение целевой функции $f(x^p)$ в узле x^p, тем больше значение $Pr(x^p,\tilde{x}^{best})$);

– вычислить вектор средних значений $\tilde{m}^p=(\tilde{m}_1^p,\tilde{m}_2^p,...,\tilde{m}_n^p)^T$, где $\tilde{m}_i^p=\dfrac{x_i^p+\tilde{x}_i^{best}}{2}$, $i=1,...,n$;

– вычислить вектор минимальных расстояний по каждой координате $\xi=(\xi_1,\xi_2,...,\xi_n)^T$, где каждая компонента ξ_i определяется по формуле:

$$\xi_i=\begin{cases}\dfrac{\Delta_i}{4}, & \text{если } c<0{,}15\%C;\\ \dfrac{\Delta_i}{8}, & \text{если } 0{,}15\%C\leq c<0{,}3\%C;\\ \dfrac{\Delta_i}{16}, & \text{если } 0{,}3\%C\leq c<0{,}6\%C;\\ \dfrac{\Delta_i}{50}, & \text{если } 0{,}6\%C\leq c<0{,}8\%C;\\ \dfrac{\Delta_i}{100}, & \text{если } c\geq 0{,}8\%C,\end{cases}$$

– вычислить координаты нового узла x^{loc}:

$$x_i^{loc}=\begin{cases}\tilde{m}_i^p, & \text{если } \left|\tilde{m}_i^p-\tilde{x}_i^{best}\right|>\xi_i \text{ и } R[0,1]\leq Pr(x^p,\tilde{x}^{best});\\ \tilde{x}_i^{best}+R[-\xi_i,\xi_i], & \text{если } \left|\tilde{m}_i^p-\tilde{x}_i^{best}\right|\leq\xi_i;\\ R\left[x_i^p,\tilde{m}_i^p\right] \text{ или } R\left[\tilde{m}_i^p,x_i^p\right] & \text{в остальных случаях,}\end{cases}$$

где $R[x,y]$ – равномерное распределение на отрезке $[x,y]$, $i=1,...,n$.

В результате шага 4 генерируется множество новых узлов $I_{loc}=\{x^{loc,1},x^{loc,2},..,x^{loc,Z}\}$, состоящее из Z узлов.

Шаг 5. *Глобальное расширение сетки*. Генерация новых узлов по направлению к глобальному экстремуму. Для каждого узла сетки с номером p, кроме наилучшего узла x^{best}, выполнить:

– вычислить фактор близости текущего узла x^p и наилучшего узла x^{best}: $Pr(x^p, x^{best}) = \dfrac{1}{1 + \left| f(x^p) - f(x^{best}) \right|}$;

– вычислить вектор средних значений $m^p = (m_1^p, m_2^p, .., m_n^p)^T$, где $m_i^p = \dfrac{x_i^p + x_i^{best}}{2}$, $i = 1, ..., n$;

– вычислить координаты нового узла x^{glob}:

$$x_i^{glob} = \begin{cases} m_i^p, & \text{если } R[0,1] \le Pr(x^p, x^{best}); \\ R\left[m_i^p, x_i^{best}\right] \text{ или } R\left[x_i^{best}, m_i^p\right], & \text{в остальных случаях,} \end{cases} \quad i = 1, ..., n.$$

В результате шага 5 генерируется множество новых узлов $I_{glob} = \{x^{glob,1}, x^{glob,2}, ..., x^{glob,P-1}\}$, состоящее из $(P-1)$ узлов.

Шаг 6. *Дополнительное расширение сетки*. Генерация новых узлов для границ сетки. Если $Z + P - 1 < T$, то выполнить шаги 6.1–6.5, иначе перейти к шагу 7.

Шаг 6.1. Найти расстояния от центра x^m множества допустимых решений до всех узлов сетки: $d(x^m, x^j) = \sqrt{\sum_{i=1}^{n} \left(x_i^m - x_i^j\right)^2}$, $j = 1, 2, ..., P$. Упорядочить все узлы по возрастанию расстояния от центра множества допустимых решений (первый элемент соответствует минимальному расстоянию, последний – максимальному): $I^{(\cdot)} = \{x^{(1)}, x^{(2)}, ..., x^{(P)}\}$, где узлы $x^{(1)}$ и $x^{(P)}$ удовлетворяют условиям: $d(x^m, x^{(1)}) = \min_{j=1,...,P} d(x^m, x^{(j)})$, $d(x^m, x^{(P)}) = \max_{j=1,...,P} d(x^m, x^{(j)})$.

Шаг 6.2. Вычислить вектор смещения $w = (w_1, w_2, ..., w_n)^T$, где $w_i = (w_i^0 - w_i^1) \cdot \dfrac{C - c}{C} + w_i^1$, $w_i^0 = \dfrac{\Delta_i}{10}$, $w_i^1 = \dfrac{\Delta_i}{100}$, $i = 1, ..., n$.

Шаг 6.3. Вычислить количество генерируемых узлов $Y = T - (Z + P - 1)$. Если $Y > P$, то положить $Y = P$.

Шаг 6.4. Генерировать по одному новому узлу для $\left[\dfrac{Y}{2}\right]$ последних узлов из упорядоченной популяции $I^{(\cdot)}$: $x^{ef,j} = ({x_1}^{ef,j}, {x_2}^{ef,j}, ..., {x_n}^{ef,j})^T$, $j = P, P-1, ..., P - \left[\dfrac{Y}{2}\right] + 1$,

$$x_i^{ef,j} = \begin{cases} x_i^{(j)} + w_i, & \text{если } x_i^{(j)} \ge 0; \\ x_i^{(j)} - w_i, & \text{если } x_i^{(j)} < 0, \end{cases} \quad i = 1, ..., n.$$

Шаг 6.5. Генерировать по одному новому узлу для $Y - \left[\dfrac{Y}{2}\right]$ первых узлов из упорядоченной популяции $I^{(\cdot)}$: $x^{if,j} = ({x_1}^{if,j}, {x_2}^{if,j}, ..., {x_n}^{if,j})^T$, $j = 1, 2, ..., Y - \left[\dfrac{Y}{2}\right]$,

$$x_i^{if,j} = \begin{cases} \left| x_i^{(j)} + w_i \right|, & \text{если } x_i^{(j)} > 0; \\ \left| x_i^{(j)} - w_i \right|, & \text{если } x_i^{(j)} \le 0, \end{cases} \quad i = 1, ..., n.$$

В результате шага 6 генерируется множество новых узлов $I_{ef,if} = \{x^{ef,1}, ..., x^{ef,\left[\frac{Y}{2}\right]}, x^{if,1}, ..., x^{if,Y-\left[\frac{Y}{2}\right]}\}$, состоящее из Y узлов.

Шаг 7. *Добавление сгенерированных узлов в текущую популяцию.*

Шаг 7.1. Вычислить значения целевой функции в узлах множеств I_{loc}, I_{glob} и $I_{ef,if}$; положить $c = c + Z + P - 1 + Y$

Шаг 7.2. Добавить узлы множеств I_{loc}, I_{glob} и $I_{ef,if}$ в текущую популяцию I_t и упорядочить узлы в популяции по убыванию соответствующих им значений целевой функции. В результате получается упорядоченная популяция, состоящая из $\tilde{P} = 2P + Z + Y - 1$ узлов: $I_t = \{x^p = (x_1^p, x_2^p, ..., x_n^p)^T,\ p = 1, 2, ..., \tilde{P}\}$, $f(x^1) = \max\limits_{p=1,...,\tilde{P}} f(x^p)$, $f(x^{\tilde{P}}) = \min\limits_{p=1,...,\tilde{P}} f(x^p)$.

Шаг 8. *Сокращение популяции.* Задать индекс узла $j = 1$. Для j-го узла сетки выполнить шаги 8.1– 8.3.

Шаг 8.1. Вычислить векторы разностей координат узла x^j и всех последующих в упорядоченной популяции I_t узлов x^k: $\delta^{j,k} = \{\delta_1^{j,k}, \delta_2^{j,k}, ..., \delta_n^{i,k}\}$, где $\delta_i^{j,k} = \left| x_i^j - x_i^k \right|$, $k = j+1, 2, ..., \tilde{P}$.

Шаг 8.2. Сравнить координаты векторов $\delta^{j,k}$, $k = j+1, 2, ..., \tilde{P}$, с координатами вектора ξ, найденного на шаге 4:

– если для k-го узла для всех $i = 1, 2, ..., n$ выполняется условие $\delta_i^{j,k} \geq \xi_i$, то узел с индексом k остается в сетке;

– если условие $\delta_i^{j,k} \geq \xi_i$ не выполняется хотя бы для одной координаты i, $i = 1, 2, ..., n$, k-го узла, то удалить из сетки узел с индексом k, положить $\tilde{P} = \tilde{P} - 1$.

Шаг 8.3. Положить $j = j + 1$. Если $j \geq \tilde{P}$, то процесс сокращения завершить; если $j < \tilde{P}$, то повторить шаги 8.1– 8.3.

В результате шага 8 остается популяция, содержащая B узлов.

Шаг 9. *Генерация новой популяции.*

Сравнить количество оставшихся узлов в сетке B с количеством узлов в начальной сетке P:

– если $B < P$, то генерировать $P - B$ новых узлов, используя равномерное распределение на множестве допустимых решений D;

– если $B \geq P$, то выбрать P первых узлов сетки, которым соответствует наибольшее значение целевой функции.

Положить $t = t + 1$.

Шаг 10. *Проверка условий окончания работы алгоритма.*

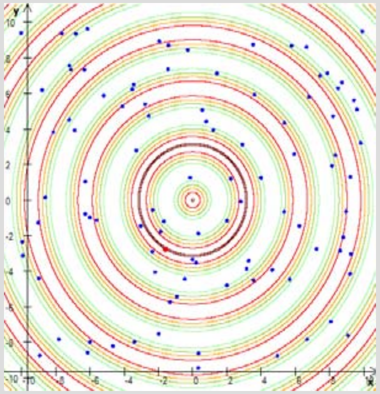
Если $c < C$, то перейти к шагу 3; иначе перейти к шагу 11.

Шаг 11. *Получение приближенного решения.*

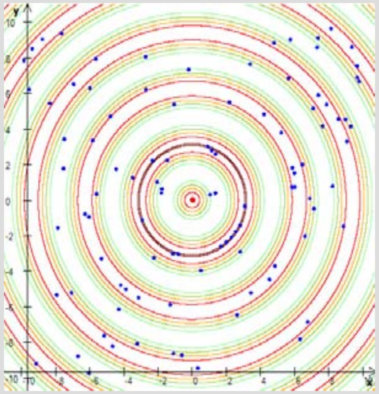
Завершить процесс поиска, в качестве приближенного решения задачи (5.1) выбрать из последней популяции узел, которому соответствует наибольшее значение целевой функции: $x^* \cong \tilde{x}^* = \arg\max\limits_{p=1,...,P} f(x^p)$.

Пример 1. Рассмотрим функцию Шаффера (табл. П.1). Зададим множество допустимых решений $x, y \in [-10;10]$. Выберем следующие параметры алгоритма: размер популяции $P = 100$; количество вычислений целевой функции $C = 5000$; количество соседних узлов $K = 5$; количество генерируемых узлов при расширении $T = 200$;

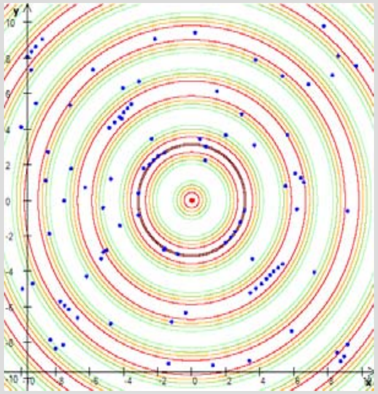
На рис. представлена начальная ($t = 0$), промежуточные ($t = 3$, $t = 15$) и конечная ($t = 21$) популяции. Результаты работы метода динамических сеток: сформировано популяций $t = 21$; узел с наилучшим значением целевой функции $(x^*; y^*)^T = (-0,0199; -0,0141)^T$; наилучшее значение целевой функции $f(x^*; y^*) = 0,9994$; отклонение от точного решения $\Delta = 0,0006$. В последней популяции узел с наилучшим значением целевой функции находится в точке глобального экстремума.



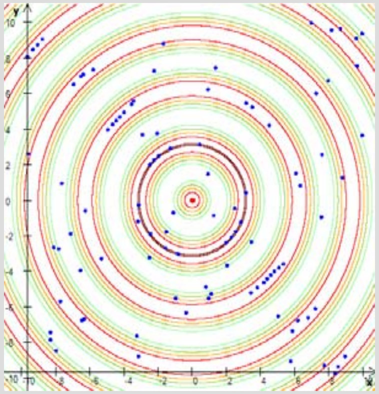
$t = 0$



$t = 3$

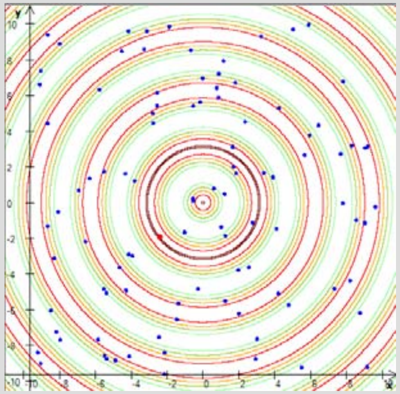


$t = 15$

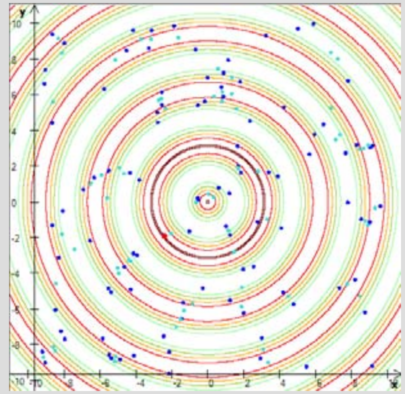


$t = 21$

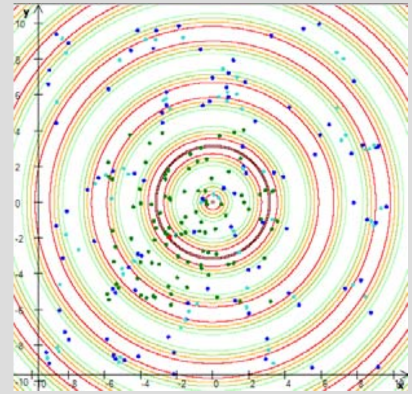
Начальная, промежуточные и конечная популяции



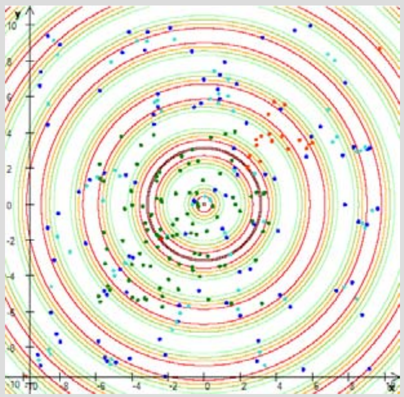
а) Начальная сетка



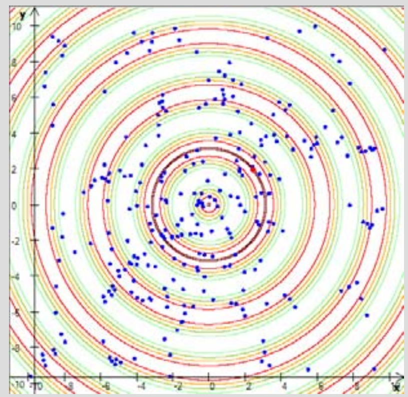
б) Локальное расширение



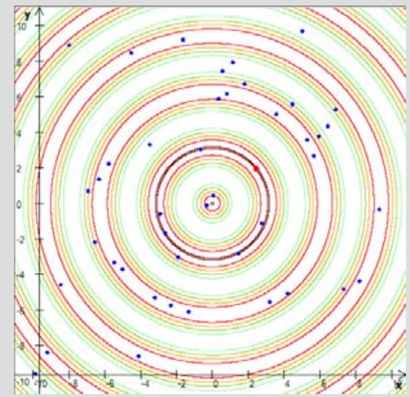
в) Глобальное расширение



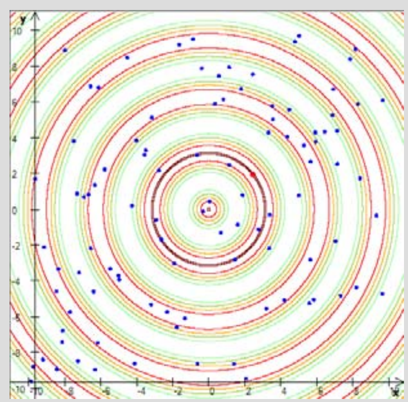
г) Дополнительное расширение



д) Объединенная сетка



е) Сокращенная сетка



ж) Новая сетка

Формирование новой сетки

I. Е. МЕТОДЫ ДИФФЕРЕНЦИАЛЬНОЙ ЭВОЛЮЦИИ

Постановка задачи

Требуется найти глобальный условный минимум функции $f(x)$ на множестве D, т.е. такую точку $x^* \in D$, что $f(x^*) = \min\limits_{x \in D} f(x)$, где $x = (x_1, x_2, ..., x_n)^T$, $D = \{x \mid x_i \in [a_i, b_i], i = 1, 2, ..., n\}$.

Стратегия поиска решения

Метод дифференциальной эволюции основан на анализе эволюционных процессов. Особенностью алгоритмов дифференциальной эволюции является использование различий между индивидами (значениями аргумента целевой функции), реализованное линейным оператором, называемым ***«дифференциация».***

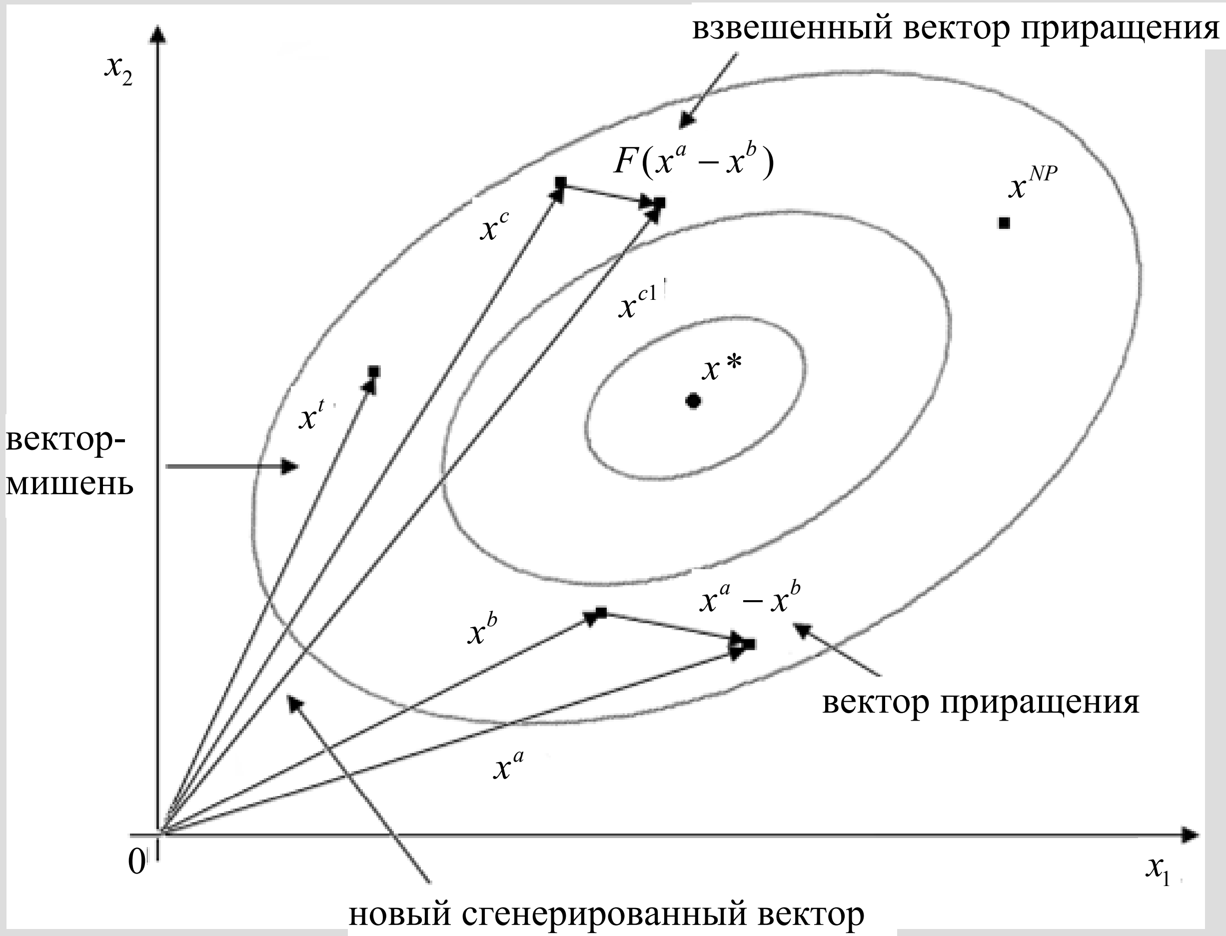
Сначала в области допустимых решений генерируется конечный набор $I_0 = \{x^k, k = 1, 2, ..., NP\} \subset D$ векторов, называемый начальной популяцией, где NP – размер популяции. Как правило, для этого используется равномерное распределение на множестве D. Далее идет циклический процесс замены текущей популяции новой. Он заканчивается, когда количество сформированных популяций оказывается равным заданному максимальному числу популяций.

Для формирования новой популяции проводится NP испытаний. При этом последовательно выбирается каждый элемент текущей популяции (он называется *вектором-мишенью* x^t), и принимается решение, остается ли он в новой популяции или нет.

Пусть в начальной популяции выбран вектор-мишень $x^t = x^1$. Среди остальных ее членов случайным образом выбираются три вектора, отличных друг от друга и от вектора-мишени: x^a, x^b, x^c. Пара векторов x^a, x^b определяет вектор приращения $x^a - x^b$. Этот вектор, умноженный на масштабирующую постоянную F (весовой коэффициент), складывается с вектором x^c. В результате данной операции, называемой *мутацией*, получается новый вектор x^{c1}:

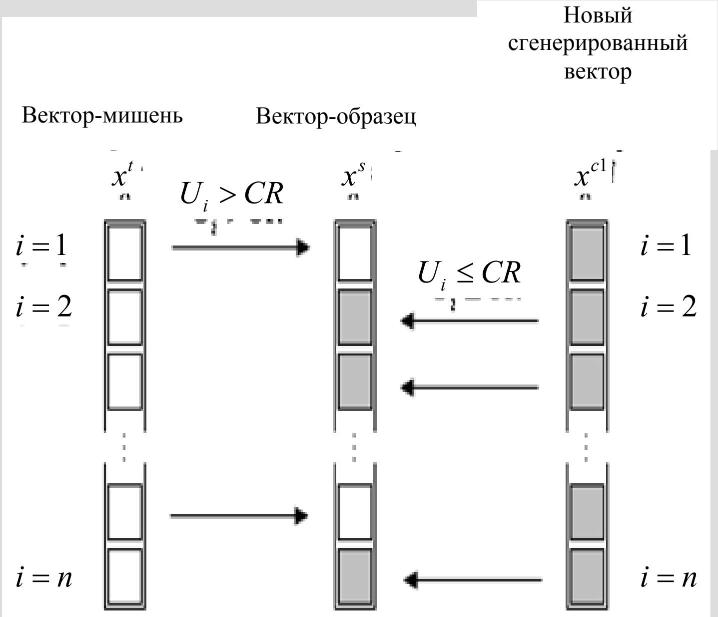
$$x^{c1} = x^c + F(x^a - x^b).$$

Если какая-либо из координат (например, с номером s) сформированного вектора x^{c1} вышла за пределы множества допустимых решений, она заменяется сгенерированной при помощи равномерного распределения на отрезке $[a_s, b_s]$ случайной величиной.



Формирование вектора x^{c1}

Далее вектор-мишень x^t и вектор x^{c1} подвергаются операции *скрещивания*. Результатом скрещивания является вектор x^s, называемый *вектором-образцом*. При этом используется параметр операции скрещивания CR. Каждая координата вектора-образца x^s совпадает либо с соответствующей координатой вектора-мишени x^t, либо с соответствующей координатой сгенерированного вектора x^{c1}. Операция скрещивания производится следующим образом: если при фиксированном номере i случайное число U_i, равномерно распределенное на отрезке $[0,1]$, больше CR, то берется координата вектора-мишени x^t, иначе – координата вектора x^{c1}. Последняя координата вектора x^s (при $i = n$) всегда определяется координатой сгенерированного вектора x^{c1}. Это делается для того, чтобы даже при $CR = 0$ вектор x^s отличался от вектора-мишени x^t хотя бы одной (последней координатой). Заметим, что при $CR = 1$ вектор-образец полностью совпадает со сгенерированным вектором x^{c1}.



Результат скрещивания

Для *формирования новой популяции* сравниваются значения целевой функции для вектора-мишени x^t и вектора-образца x^s. Если значение меньше для вектора-образца, то он замещает в популяции вектор-мишень, иначе вектор-мишень остается.

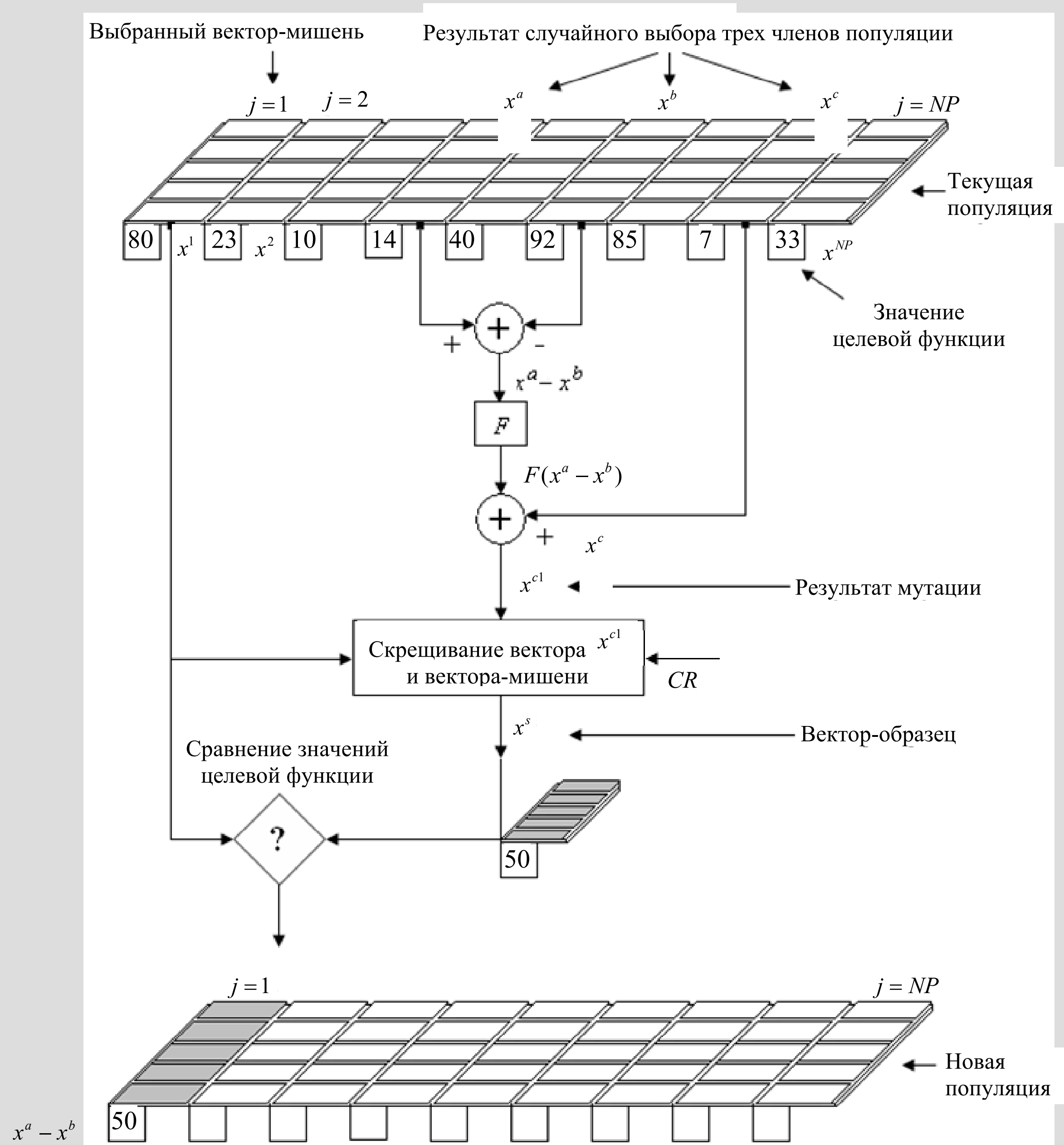


Схема метода дифференциальной эволюции

Алгоритм решения задачи

Шаг 1. Задать размер популяции NP; весовой коэффициент F; параметр операции скрещивания CR; максимальное количество популяций M.

Шаг 2. Формирование начальной популяции. Сгенерировать на множестве D допустимых решений NP векторов: $x^1, x^2, ..., x^{NP}$. Положить число популяций $m = 0$.

Шаг 3. Положить $j = 1$ (вектором-мишенью выбран вектор $x^t = x^1$).

Шаг 4. Из числа оставшихся членов текущей популяции I_m случайным образом выбрать три различных члена: x^a, x^b, x^c, отличающихся от вектора-мишени x^t.

Шаг 5. Сформировать вектор x^{c1}: $x^{c1} = x^c + F(x^a - x^b)$. Если в полученном векторе x^{c1} s-я координата $x_s^{c1} \notin [a_s, b_s]$, то сгенерировать с помощью равномерного распределения на отрезке $[a_s, b_s]$ случайную величину и x_s^{c1} приравнять ей.

Шаг 6. Сформировать вектор-образец x^s.

Для этого положить $i = 1$ и выполнить следующие действия:

а) сгенерировать при помощи равномерного распределения на [0;1] случайное число U_i;

б) найти x_i^s:

– если $i = n$, то $x_i^s = x_i^{c1}$;

– если $U_i \le CR$, то $x_i^s = x_i^{c1}$;

– если $U_i > CR$, то $x_i^s = x_i^t$;

в) проверить выполнение неравенства $i \ge n$:

– если неравенство выполнено, то процедуру формирования вектора x^s завершить;

– если нет, положить $i = i + 1$ и перейти к шагу 6а).

Шаг 7. Формирование новой популяции. Вычислить значения целевой функции для вектора-образца x^s и вектора-мишени x^t. Сравнить величины $f(x^s)$ и $f(x^t)$:

а) если $f(x^s) < f(x^t)$, то поместить вектор-образец x^s в новую популяцию;

б) если $f(x^s) \geq f(x^t)$, то поместить вектор-мишень x^t в новую популяцию.

Шаг 8. Проверить выполнение неравенства $j \geq NP$:

а) если неравенство выполнено, то переход к новой популяции завершен, перейти к шагу 9;

б) если нет, то положить $j = j+1$, $x^t = x^j$ (вектором-мишенью выбран вектор $x^t = x^j$) и перейти к шагу 4.

Шаг 9. Проверить условие окончания:

а) если число сформированных популяций $m < M$, то положить $m = m+1$ и перейти к шагу 3;

б) если $m = M$, то процесс закончить. Выбрать из последней популяции вектор с наименьшим значением целевой функции и считать его приближенным решением задачи.

Модификации алгоритма

Другие варианты линейного оператора дифференциации, используемого на шаге 5 алгоритма:

а) $x^{c1} = x^e + F(x^a + x^b - x^c - x^d)$,

где вместо трех элементов популяции выбираются пять;

б) $x^{c1} = x_{best} + F(x^a - x^b)$,

где вместо x^c выбирается элемент x_{best} популяции, которому соответствует наилучшее значение целевой функции;

в) $x^{c1} = x_{best} + F(x^a + x^b - x^c - x^d)$;

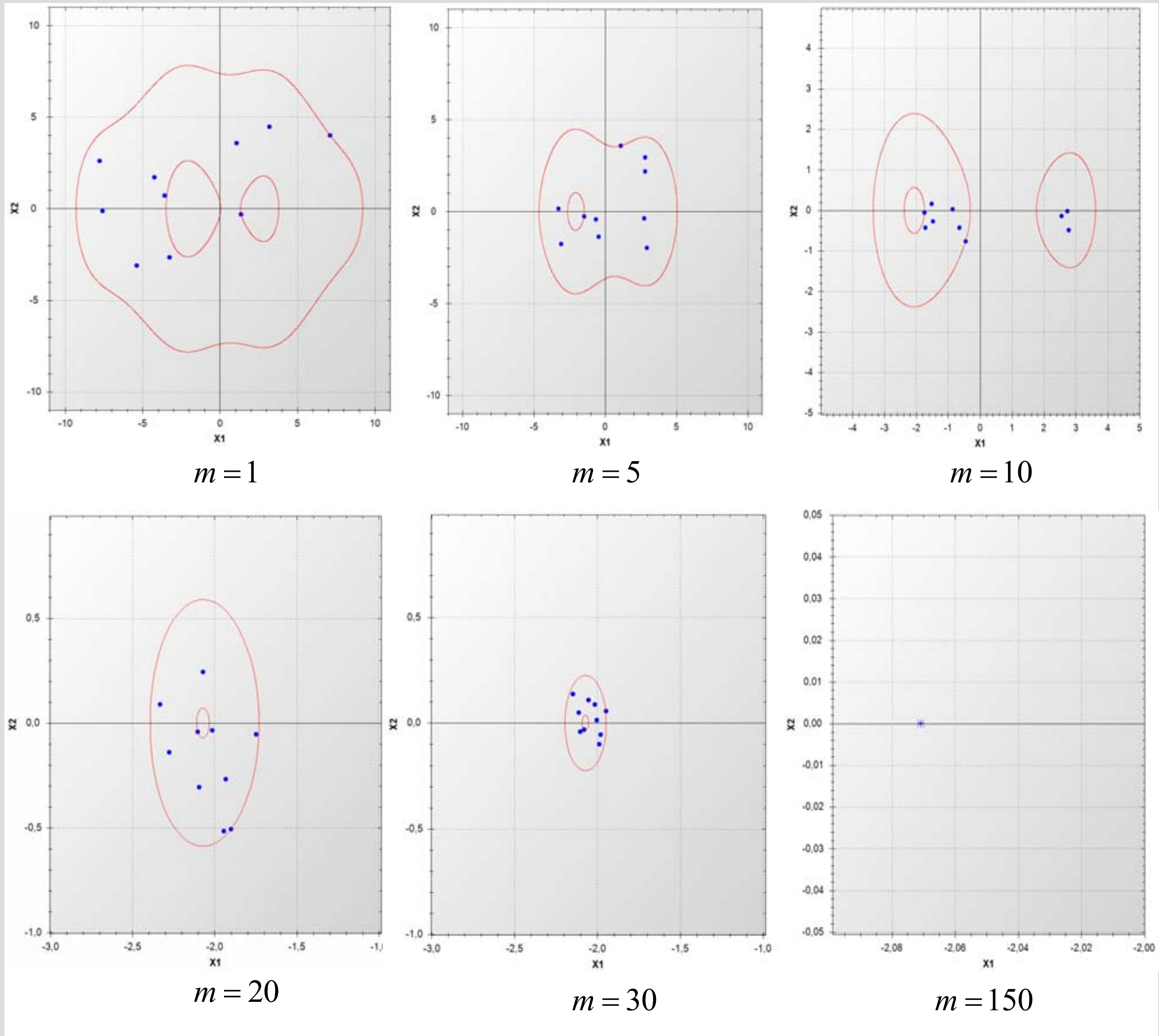
г) $x^{c1} = x^t + \lambda(x_{best} - x^a) + F(x^b - x^c)$, где x^t – вектор-мишень; λ, F – коэффициенты.

Обобщение известных схем в форме

$$x^{c1} = \beta + F\,\delta,$$

где β – базисный вектор, δ – вектор различий (направление поиска), F – коэффициент влияния.

Пример. Рассмотрим двухэкстремальную функцию (табл. П.1). Зададим множество допустимых решений $x, y \in [-10;10]$. Выберем следующие параметры метода дифференциальной эволюции: размер популяции $NP = 10$, весовой коэффициент $F = 0.8$,параметр операции скрещивания $CR = 0.9$, максимальное количество популяций $M = 150$. Результаты работы метода дифференциальной эволюции: особь с наилучшей приспособленностью $(x^*; y^*)^T = (-2,07;0)^T$; наилучшая приспособленность $f(x^*; y^*) = -\ 6,489$; отклонение от точного решения $\Delta = 0,0008$.■



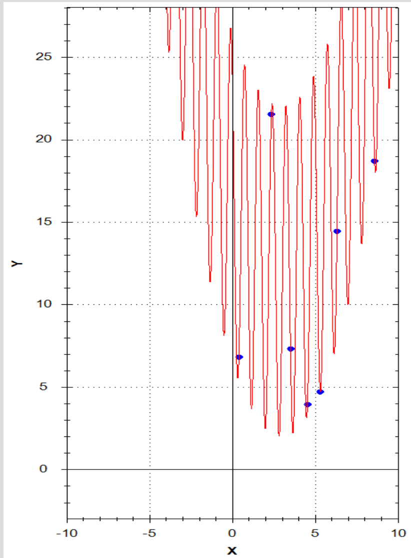
$m = 1$ $m = 5$ $m = 10$

$m = 20$ $m = 30$ $m = 150$

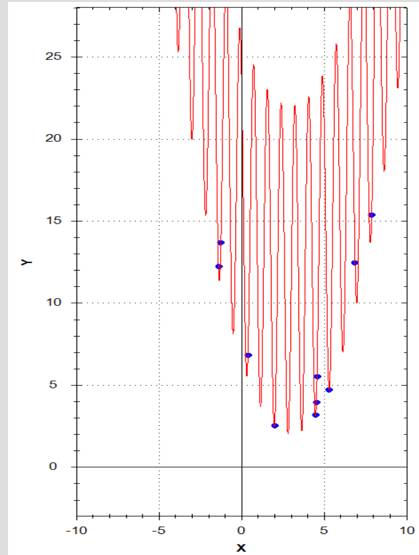
Начальная, промежуточные и конечная популяции

Пример. Рассмотрим функцию одной переменной: $f(x) = \gamma(x) + O^{K_2,H}(x)$,

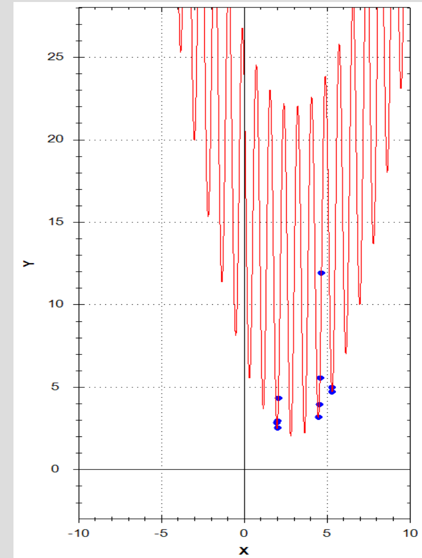
где $\gamma(x) = 0,5(x - c_2)^2 + 2$, $O^{K_2,H}(x) = -H\cos\left(\pi\dfrac{K_2}{5}(x - c_1)\right) + H$. Параметры: $H = 10$, $K_2 = 12$, $c_1 = -3$, $c_2 = 3$, $x \in [-10;10]$, $NP = 10$, $F = 0.8$, $CR = 0.9$, $M = 50$. Результаты: $x^* = 2,833$; $f(x^*) = 2,013$.



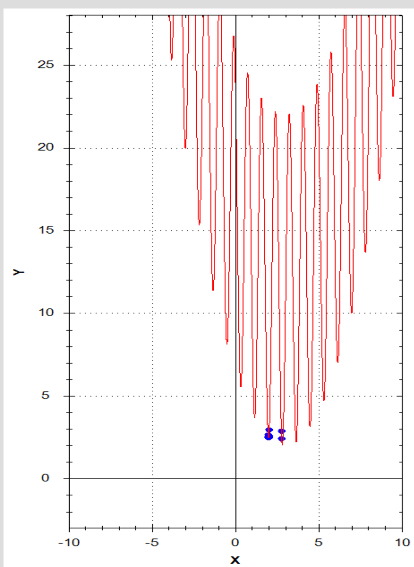
$m = 1$



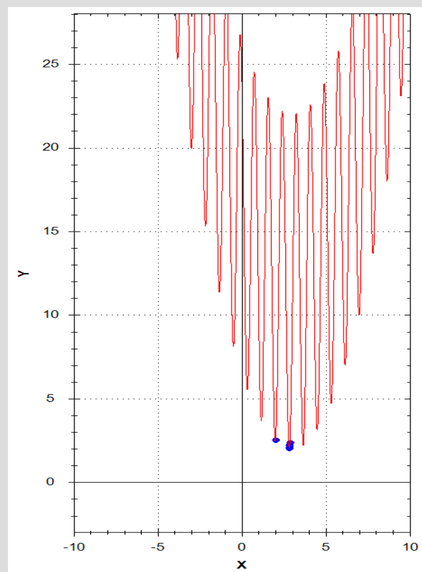
$m = 5$



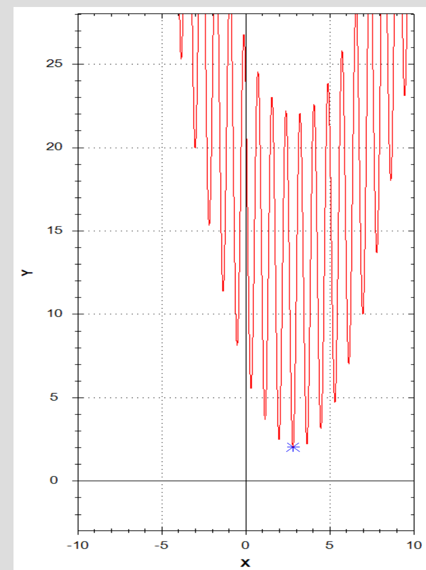
$m = 10$



$m = 20$



$m = 30$



$m = 50$

Начальная, промежуточные и конечная популяции