Лабораторная работа №7

Оптимизация функций многих переменных с помощью роевых алгоритмов

Цель работы: оптимизация функций многих переменных методом роевого интеллекта. Графическое отображение результатов оптимизации.

Основной роевой алгоритм

Роевой алгоритм (РА) использует рой частиц, где каждая частица представляет потенциальное решение проблемы.

Поведение частицы в гиперпространстве поиска решения все время подстраивается в соответствии со своим опытом и опытом своих соседей.

Кроме этого, каждая частица помнит свою лучшую позицию с достигнутым локальным лучшим значением целевой (фитнесс-) функции и знает наилучшую позицию частиц - своих соседей, где достигнут глобальный на текущий момент оптимум.

В процессе поиска частицы роя обмениваются информацией о достигнутых лучших результатах и изменяют свои позиции и скорости по определенным правилам на основе имеющейся на текущий момент информации о локальных и глобальных достижениях.

При этом глобальный лучший результат известен всем частицам и немедленно корректируется в том случае, когда некоторая частица роя находит лучшую позицию с результатом, превосходящим текущий глобальный оптимум.

Каждая частица сохраняет значения координат своей траектории с соответствующими лучшими значениями целевой функции, которые обозначим y_i, которая отражает когнитивную компоненту.

Аналогично значение глобального оптимума, достигнутого частицами роя, будем обозначать $\hat{y}_i$, которое отражает социальную компоненту.

Каждая i-я частица характеризуется в момент времени t своей позицией $x_i(t)$ в гиперпространстве и скоростью движения $v_i(t)$.

Позиция частицы изменяется в соответствии со следующей формулой:

$x_i(t+1) = x_i(t) + v_i(t+1)$, где $x_i(0) \sim (x_{min}, x_{max})$.

(11.1)

Вектор скорости $v_i(t+1)$ управляет процессом поиска решения и его компоненты определяются с учетом когнитивной и социальной составляющей следующим образом:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)\left[y_{ij}(t) - x_{ij}(t)\right] + c_2 r_{2j}(t)\left[\hat{y}_j(t) - x_{ij}(t)\right] \qquad (11.2)$$

Здесь: $v_{ij}(t)$ - j-ая компонента скорости ($j = 1,\ldots,n_x$) i-ой частицы в момент времени t , $x_{ij}(t)$- j-я координата позиции i-й частицы , c_1 и c_2 – положительные коэффициенты ускорения (часто полагаемые 2), регулирующие вклад когнитивной и социальной компонент, $r_{1j}(t), r_{2j}(t) \sim (0,1)$ - случайные числа из диапазона [0,1], которые генерируются в соответствии с нормальным распределением и вносят элемент случайности в процесс поиска. Кроме этого $y_{ij}(t)$- персональная лучшая позиция по j-й координате i-ой частицы, а $\hat{y}_j(t)$ –лучшая глобальная позиция роя, где целевая функция имеет экстремальное значение.

При решении задач минимизации персональная лучшая позиция в следующий момент времени (t+1) определяется следующим образом:

$$y_i(t+1) = \begin{cases} y_i(t)\ if f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) if f(x_i(t+1)) < f(y_i(t)) \end{cases}$$

(11.3)

где $f: R^{n_\infty} \to R$ - фитнесс-функция.

Как и в эволюционных алгоритмах фитнесс-функция измеряет близость текущего решения к оптимуму.

Глобальная лучшая позиция $\hat{y}_j(t)$ в момент t определяется в соответствии с

$$\hat{y}_j(t) \epsilon \{y_0(t), \dots, y_{n_s}(t)\} | f\left(\hat{y}_j(t)\right) =$$

$$\min\left\{f\left(y_0(t)\right), \dots, f\left(y_{n_s}(t)\right)\right\}, \qquad (11.4)$$

Где n_s – общее число частиц в рое.

В процессе поиска решения описанные действия выполняются для каждой частицы роя. Укрупненный основной роевой алгоритм представлен ниже.

Глобальный роевой

Создание инициализация n_x-мерного роя;
repeat
for каждой частицы i=1,…,n_s **do**
// определить персональную лучшую позицию
If $f(x_i) < f(y_i)$ **then**
$y_i = x_i$;
end
// определить глобальную лучшую позицию
if $f(y_i) < f(\hat{y})$ **then**
$(\hat{y}) = y_i$;
end
end
for каждой частицы i=1,…,n_s **do**
коррекция скорости согласно (11.2);
коррекция позиции согласно (11.1);
end
until критерий останова не выполнен;

Рассмотрим влияние различных составляющих при вычислении скорости частицы в соответствии с (11.2).

Первое слагаемое в (11.2) $v_i(t)$сохраняет предыдущее направление скорости i-й частицы и может рассматриваться как момент, который препятствует резкому изменению направления скорости и выступает в роли инерционной компоненты.

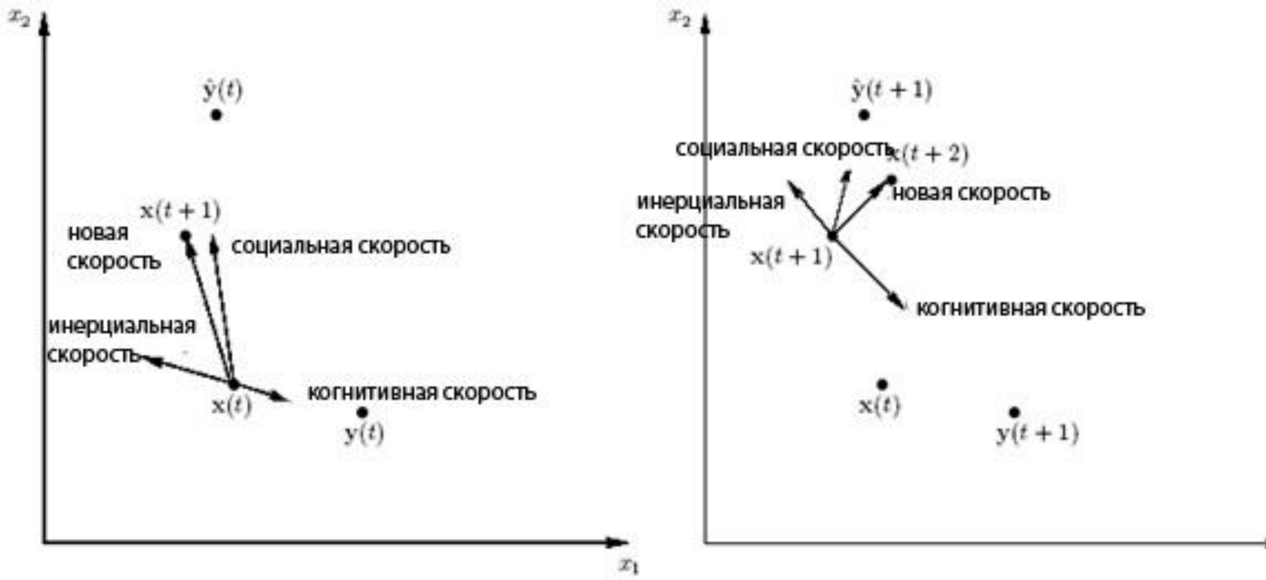
Когнитивная компонента $c_1 r_1 (y_i - x_i)$ определяет характеристики частицы относительно ее предистории, которая хранит лучшую позицию данной частицы.

Эффект этого слагаемого в том, что оно пытается вернуть частицу назад в лучшую достигнутую позицию.

Третье слагаемое $c_2 r_2(\hat{y} - x_i)$ определяет социальную компоненту, которая характеризует частицу относительно своих соседей.

Эффект социальной компоненты в том, что она пытается направить каждую частицу в сторону достигнутого роем (или его некоторым ближайшим окружением) глобального оптимума.

Графически это наглядно иллюстрируется для двумерного случая, как это показано на рис.



(a) скорость во время t (b) скорость во время t+1

Представленный основной роевой алгоритм часто называют глобальным РА (Global Best PSO), поскольку здесь при коррекции скорости частицы используется информация о положении достигнутого глобального оптимума, которая определяется на основании информации, передаваемой всеми частицами роя.

В противоположность этому подходу часто используется локальный РА, где при коррекции скорости частицы используется информация, передаваемая только в каком-то смысле ближайшими соседними частицами роя.

Тестовые примеры

Для данного вида задачи существует большое число тестовых примеров – Benchmark-ов. Для данных тестов произведено большое число исследований на скорость алгоритма, количество эпох для достижения результата и пр. С результатами этих исследований можно ознакомиться в научной литературе, доступной в Internet.

Многочисленные исследования доказывают, что РА не менее эффективны, а часто гораздо лучше справляются с задачами оптимизации в многомерных пространствах, при этом более просты в реализации из-за отсутствия процедур кодирования и декодирования хромосом.

Порядок выполнения лабораторной работы

1. Создать программу, использующую РА для нахождения оптимума функции согласно таблице вариантов, приведенной в приложении А. Для всех Benchmark-ов оптимумом является минимум. Программу выполнить на встроенном языке пакета Matlab (или любом, доступным вам, языке программирования).
2. Для n=2 вывести на экран график данной функции с указанием найденного экстремума, точек популяции. Для вывода графиков использовать стандартные возможности пакета Matlab. Предусмотреть возможность пошагового просмотра процесса поиска решения.
3. Исследовать зависимость времени поиска, числа поколений (генераций), точности нахождения решения от основных параметров генетического алгоритма:

   - число особей в популяции
   - вероятность мутации.

   Критерий остановки вычислений – повторение лучшего результата заданное количество раз или достижение популяцией определенного возраста (например, 100 эпох).

4. Повторить процесс поиска решения для n=3, n=5, n=10,сравнить результаты, скорость работы программы.

Содержание отчета.

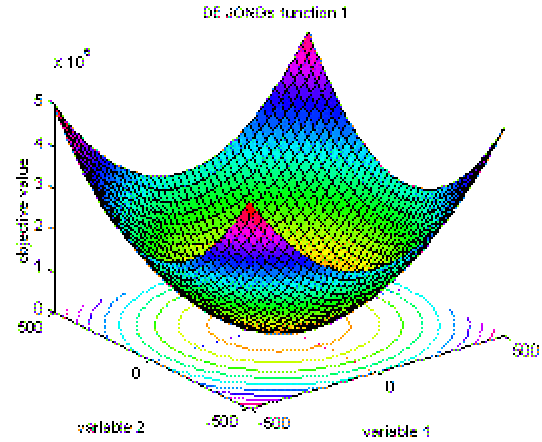
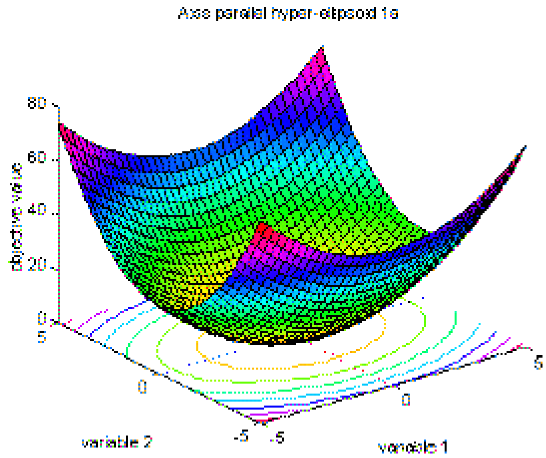
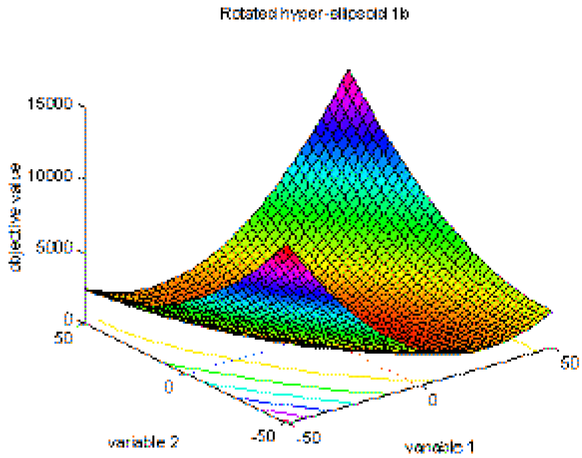
1. Титульный лист установленной формы.
2. Условие задания с вариантом.
3. Распечатанный листинг программы.
4. Распечатка результатов выполнения программы (графиков);
5. Диаграммы исследованных зависимостей.

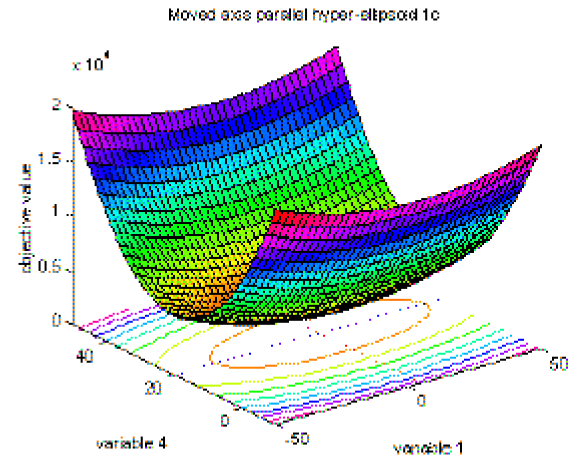
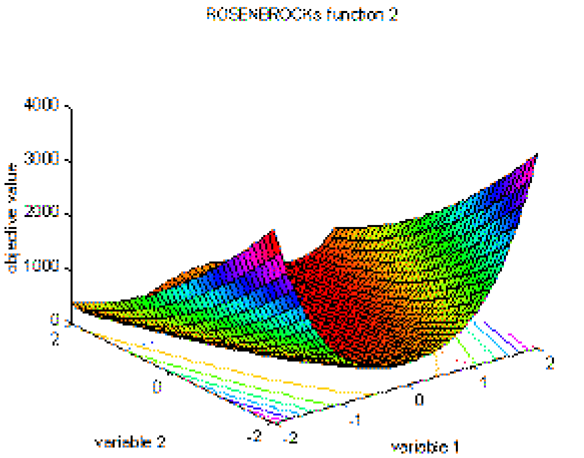
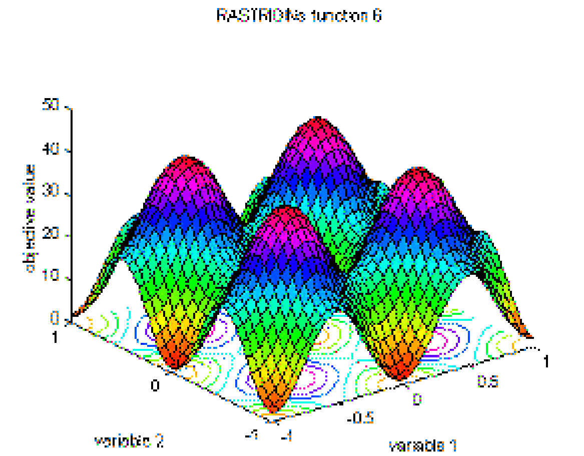
Контрольные вопросы

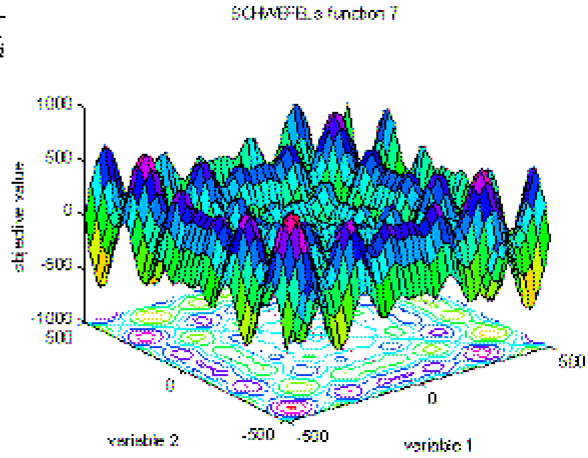
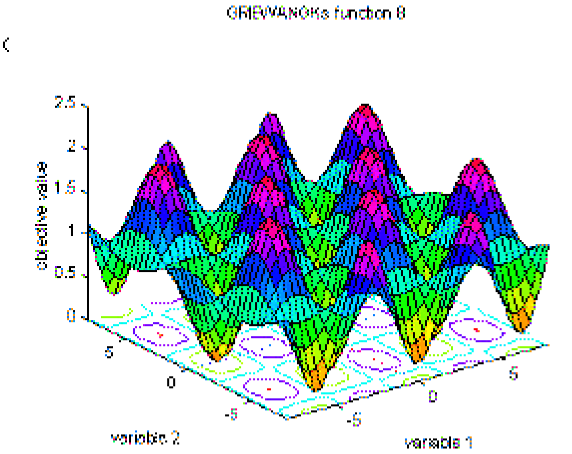
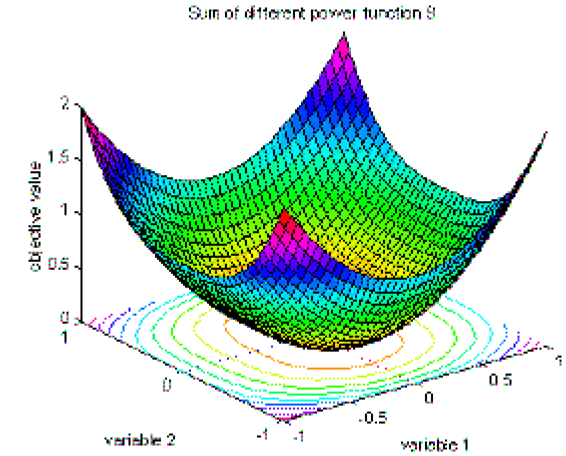
1. Как представляется потенциальное решение в РА?
2. Как производится коррекция скорости частицы?
3. Что такое когнитивная составляющая?
4. Что такое социальная составляющая?
5. Опишите глобальный РА.
6. Чем отличаются глобальный и локальный РА?
7. Какие используются социальные структуры?
8. Опишите локальный РА.
9. Определите персональную лучшую позицию.
10. Определите глобальную лучшую позицию.
11. Приведите основные аспекты РА.
12. Приведите основные условия останова РА.
13. Как выполняется инициализация в РА?
14. Приведите основные параметры РА.
15. Какие существуют модификации РА?
16. Что общего между РА, ЭС и ГА?
17. Какие различия имеют место между РА,ГА и ЭС?

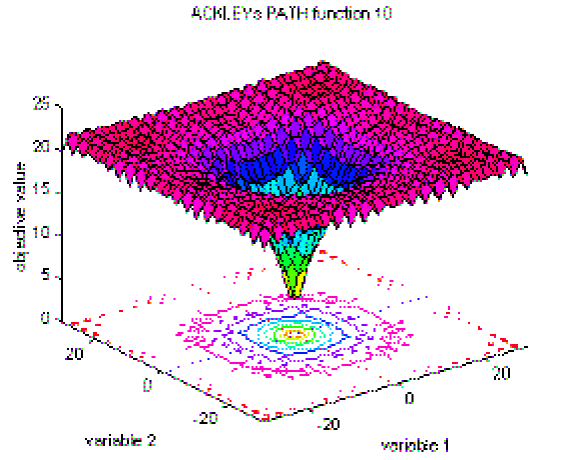
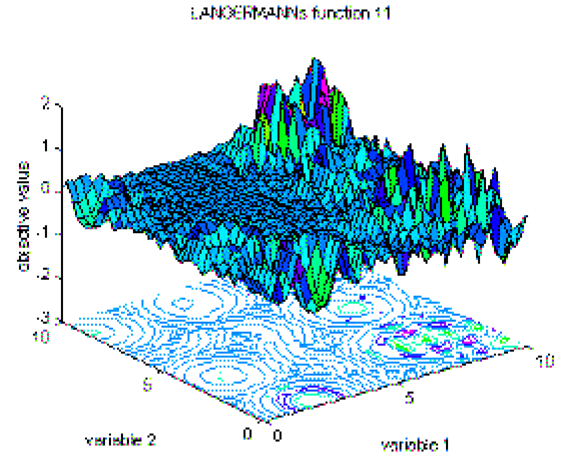
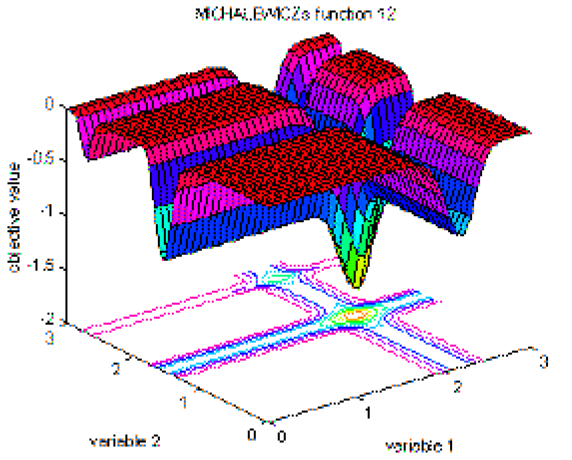
Приложение А.

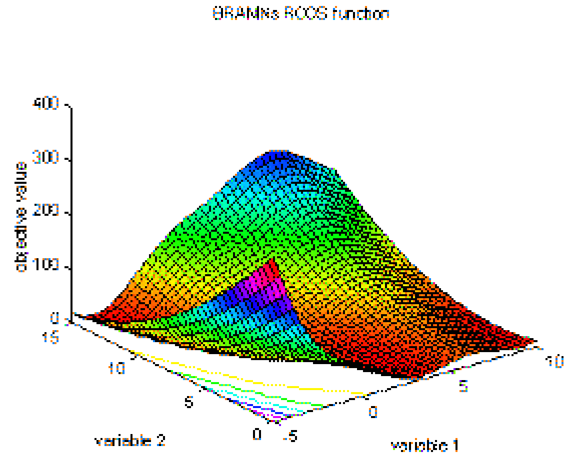
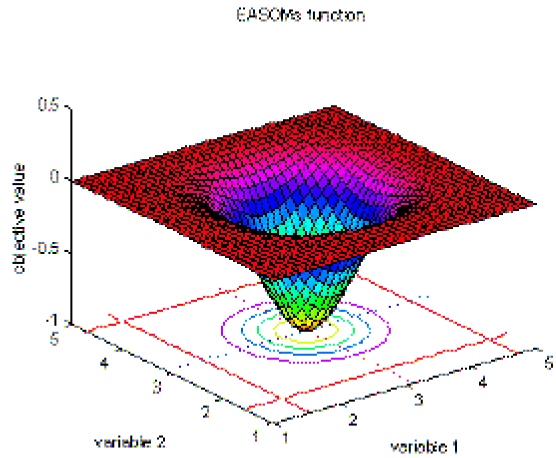
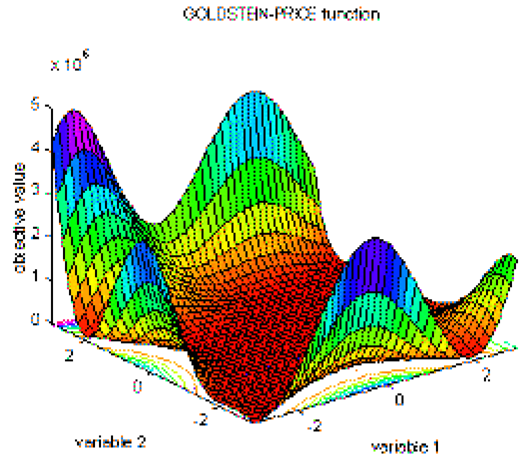
Индивидуальные задания на лабораторную работу №2.

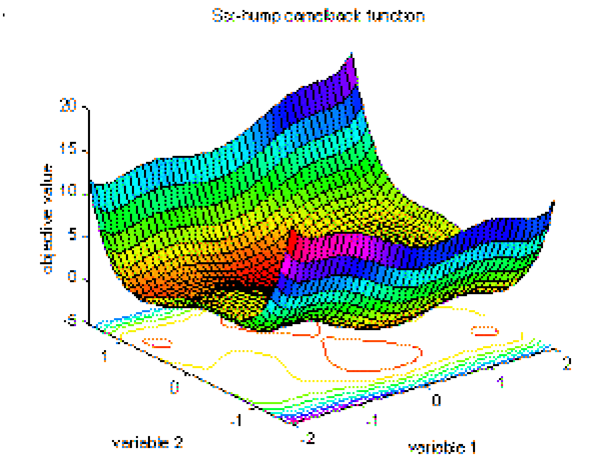
№ вв.	Название	Оптимум	Вид функции	График функции
1	De Jong's function 1	global minimum f(x)=0; x(i)=0, I=1:n.	$f_1(x)=\sum_{i=1}^{n} x_i^2$ -5.1 f1(x)=sum(x(i)^2), i=1:n;	
2	Axis parallel hyper-ellipsoid function	global minimum f(x)=0; x(i)= 0, i=1:n.	$f_{1a}(x)=\sum_{i=1}^{n} i\cdot x_i^2$ -5 f1a(x)=sum(i·x(i)^2), i=1:n;	
3	Rotated hyper-ellipsoid function	global minimum f(x)=0; x(i)=0, i=1:n	$f_{1b}(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$ f1b(x)=sum(sum(x(j)^2), j=1:i), i=1:n; .	

4	Moved axis parallel hyper-ellipsoid function	global minimum f(x)=0; x(i)= 5*i, i=1:n	$f_{1c}(x) = \sum_{i=1}^{n} 5i \cdot x_i^2$ f1c(x)=sum(5*i·x(i)^2), i=1:n;	
5	Rosenbrock's valley (De Jong's function 2)	global minimum f(x)=0; x(i)=1, i=1:n.	$f_2(x) = \sum_{i=1}^{n-1} 100 \cdot (x_{i+1} - x_i$ f2(x)=sum(100·(x(i+1)-x(i)^2)^2+(1-x(i))^2), i=1:n-1;	
6	Rastrigin's function 6	global minimum f(x)=0; x(i)=0, i=1:n.	$f_6(x) = 10 \cdot n + \sum_{i=1}^{n} (x_i^2 -$ f6(x)=10·n+sum(x(i)^2-10·cos(2·pi·x(i))), i=1:n;	

7	Schwefel's function 7	global minimum f(x)=n·418.9829; x(i)=420.9687, i=1:n.	$f_7(x)=\sum_{i=1}^{n} -x_i \cdot \sin\left(\sqrt{\lvert x_i \rvert}\right)$ f7(x)=sum(-x(i)·sin(sqrt(abs(x(i))))), i=1:n;	
8	Griewangk's function 8	global minimum f(x)=0; x(i)=0, i=1:n	$f_8(x)=\sum_{i=1}^{n}\frac{x_i^2}{4000}-\prod_{i=1}^{n} \cos$ f8(x)=sum(x(i)^2/4000)-prod(cos(x(i)/sqrt(i)))+1, i=1:n;	
9	Sum of different power function 9	global minimum f(x)=0; x(i)=0, i=1:n.	$f_9(x)=\sum_{i=1}^{n}\lvert x_i \rvert^{(i+1)}$ f9(x)=sum(abs(x(i))^(i+1)), i=1:n;	

10	Ackley's Path function 10	global minimum f(x)=0; x(i)=0, i=1:n.	$f_{10}(x) = -a \cdot e^{-b\cdot\sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}}} - e$ f10(x)=-a·exp(-b·sqrt(1/n·sum(x(i)^2)))-exp(1/n·sum(cos(c·x(i))))+a+exp(1); a=20; b=0.2; c=2·pi; i=1:n;	
11	Langermann's function 11	global minimum f(x)=-1.4 (for m=5); x(i)=???, i=1:n.	$f_{11}(x) = -\sum_{i=1}^{m} c_i \left(e^{-\frac{\|x-A(i)\|^2}{\pi}} \cdot \cos\right.$ f11(x)=-sum(c(i)·(exp(-1/pi·sum((x-A(i))^2))·cos(pi·sum((x-A(i))^2)))), i=1:m, m=5; A(i),C(i)<>0, m=5	
12	Michalewicz's function 12	global minimum f(x)=-4.687 (n=5); x(i)=???, i=1:n. f(x)=-9.66 (n=10); x(i)=???, i=1:n.	$f_{12}(x) = -\sum_{i=1}^{n} \sin(x_i) \cdot \left(\sin\right.$ f12(x)=-sum(sin(x(i))·(sin(i·x(i)^2/pi))^(2·m)), i=1:n, m=10; проверить для n=5,10	

13	Branins's rcos function	global minimum f(x1,x2)=0.397887; (x1,x2)=(-pi,12.275), (pi,2.275), (9.42478,2.475).	$f_{Bran}(x_1,x_2)=a\cdot(x_2-b\cdot x_1^2+c\cdot x_1$ $a=1,\quad b=\frac{5.1}{4\cdot\pi^2},\quad c=\frac{5}{\pi},\quad d$ fBran(x1,x2)=a·(x2-b·x1^2+c·x1-d)^2+e·(1-f)·cos(x1)+e; a=1, b=5.1/(4·pi^2), c=5/pi, d=6, e=10, f=1/(8·pi);	
14	Easom's function	global minimum f(x1,x2)=-1; (x1,x2)=(pi,pi).	$f_{Easo}(x_1,x_2)=-\cos(x_1)\cdot\cos($ fEaso(x1,x2)=-cos(x1)·cos(x2)·exp(-((x1-pi)^2+(x2-pi)^2));	
15	Goldstein-Price's function	global minimum f(x1,x2)=3; (x1,x2)=(0,-1).	$f_{Gold}(x_1,x_2)=(1+(x_1+x_2+$ $\cdot(30+(2x_1-3$ $-2\le x_i\le 2,$ fGold(x1,x2)=[1+(x1+x2+1)^2·(19-14·x1+3·x1^2-14·x2+6·x1·x2+3·x2^2)]·[30+(2·x1-3·x2)^2·(18-32·x1+12·x1^2+48·x2-36·x1·x2+27·x2^2)]	

16	Six-hump camel back function	global minimum f(x1,x2)=-1.0316; (x1,x2)=(-0.0898,0.7126), (0.0898,-0.7126).	$f_{Sixh}(x_1,x_2) = (4 - 2.1x_1^2 + x_1^{4/3})\cdot$ fSixh(x1,x2)=(4-2.1·x1^2+x1^4/3)·x1^2+x1·x2+(-4+4·x2^2)·x2^2;	

Для остальных вариантов берется строчка, соответствующая остатку от деления номера варианта на 16 (для 17-1, 18-2 и т.д.)