Цель работы: реализовать непрерывный генетический алгоритм, протестировать его на многоэкстремальных функциях.

Ход работы:

В таблице 1 приведены исходные данные тестовых функций и параметров алгоритма. Радиус оптимальности равен 0.5.

Таблица 1 – Исходные данные

Функция	Положение оптимума	Границы
Розенброка	[1 1]	[-10;10]
Гриванка	[0 0]	[-10;10]
Химмельблау	[3 2]	[-10;10]

Листинг программы:

```
clear all; clc;
%% Начальные параметры
mode = 3; %Выбор функции
Npop = 50; %Количество популяци
Nvar = 2; %Количество переменных функции
iga = 0; %Номер итерации
u = 0.2; % Процент мутаций
pop_temp = zeros(Npop,2);
Nkeep = Npop*0.5; %Количество отсеиваемых хромосом
cost = zeros(Npop,1); %вектор стоимости
beta = 0.4; %коэффициент при скрещивании
pop_new = zeros(Nkeep,Nvar); %Массив новых генов (потомков)
maxit=100; %максимальное количество итераций
nmut = ceil(u*(Npop-1)*Nvar); %Количество мутаций
if mode == 1
    fitness = @(x) 100*(x(1).^2-x(2).^2).^2+(x(1)-1).^2; %Целевая функция
elseif mode == 2
    fitness = @(x) 100*(x(1).^2-x(2).^2).^2+(x(1)-1).^2;
elseif mode == 3
    fitness = @(x) (x(1).^2+x(2)-11).^2+(x(1)+x(2).^2-7).^2; % Функция Химмельблау
end
M = ceil((Npop-Nkeep)/2); %Количество скрещиваний
%Ограничения генов
phi = [-10 -10]; %наименьшее значение доступного диапазона генов
plo = [10 10]; %наибольшее значение доступного диапазона генов

%% Формирование начальной популяции
pop = rand(Npop,Nvar); %нормализованные значения от 0 до 1
pop(:,1) = (phi(1,1)-plo(1,1))*pop(:,1)+plo(1,1);
pop(:,2) = (phi(1,2)-plo(1,2))*pop(:,2)+plo(1,2);
%массив pop - это начальная популяция, выбранная случайным образом в
%диапазоне от -5 до 5
%формирование стомости каждой хромосомы
for i=1:Npop
    cost(i) = fitness([pop(i,:)]);
end
[cost ind] = sort(cost);
%Сортировка популяций по их приспособленности к выживанию
for i=1:Npop
    pop_temp(i,:)=pop(ind(i),:);
end
```

```
pop = pop_temp;

%% Цилк программы
while iga<maxit
    iga = iga+1;
% % Формирование родителей случайным образом
for i=1:Nkeep
ma = pop(ceil(Nkeep*rand(Nkeep, 1)),:); %вектор строк в pop для parent_1
pa = pop(ceil(Nkeep*rand(Nkeep, 1)),:); %вектор строк в pop для parent_2
end

%ДОДЕЛАТЬ
%Турнирная сетка
% for i=1:Nkeep
%     cost_temp = 0;
%     numbers = randi([1,50],1,ceil(Npop*0.14));
%     for j=1:ceil(Npop*0.14)
%         cost_temp(j) = fitness([pop(numbers(j),:)]);
%     end
%     min_value = min(cost_temp);
% end
%Скрещивание родителей, всего новых особей (Npop - Nkeep)/2
for i=1:M
    for j=1:Nvar
        pop_new(i,j) = beta*ma(i,j)+(1-beta)*pa(i,j); %потомки родителей
    end
end
%Добавление потомков в популяцию
k=1;
for i=(Nkeep+1):Npop
    pop(i,:) = pop_new(k,:);
    k=k+1;
end
% %Процесс мутации
mrow=ceil(rand(1,nmut)*(Npop-1))+1;
mcol=ceil(rand(1,nmut)*Nvar);
for i=1:nmut
    pop(mrow(i),mcol(i)) = (phi(1)-plo(1))*rand+plo(1);
end

%Результаты функции
for i=1:Npop
    cost(i) = fitness([pop(i,:)]);
end
[cost ind] = sort(cost);
for i=1:Npop
    pop_temp(i,:)=pop(ind(i),:);
end
pop = pop_temp;
minc(iga)=cost(1); %Минимальная стоимость
meanc(iga)=mean(cost); %Средняя стоимость
end
```

1. Исследование разработанного генетического алгоритма (ГА) на тестовых функциях:

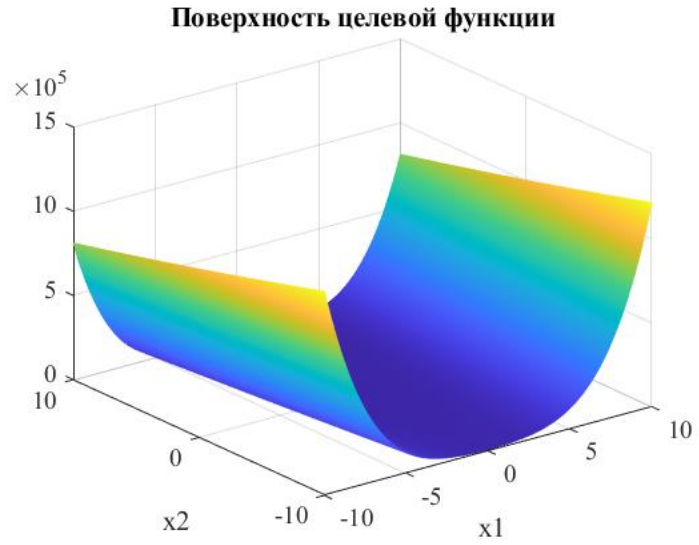

Рисунок 1 – Поверхность функции Розенброка

Рисунок 2 – График изменения популяции в среднем

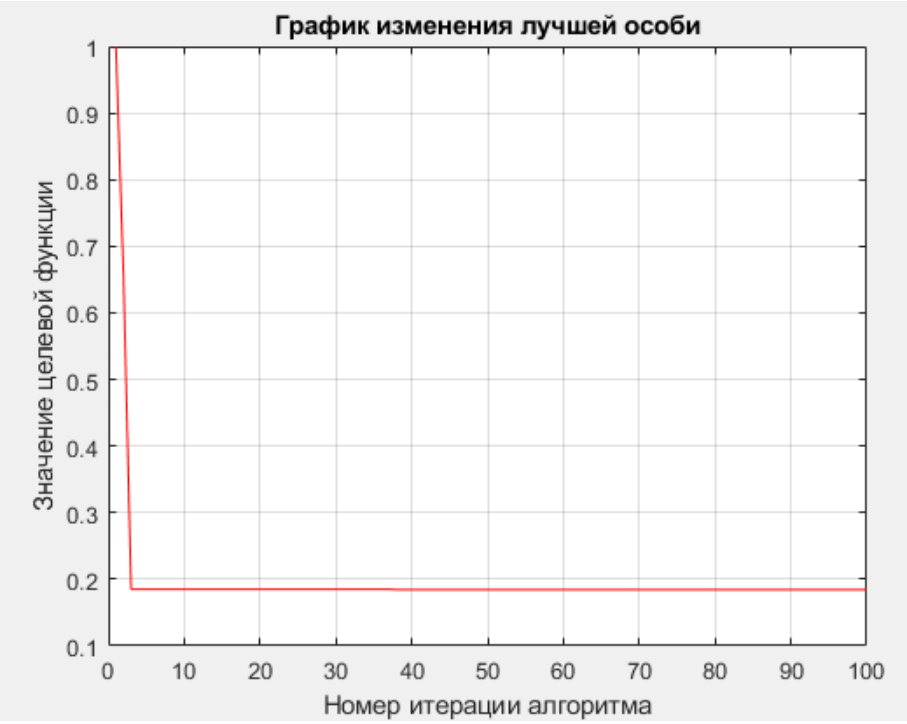

Рисунок 3 – График изменения популяции лучшей особи

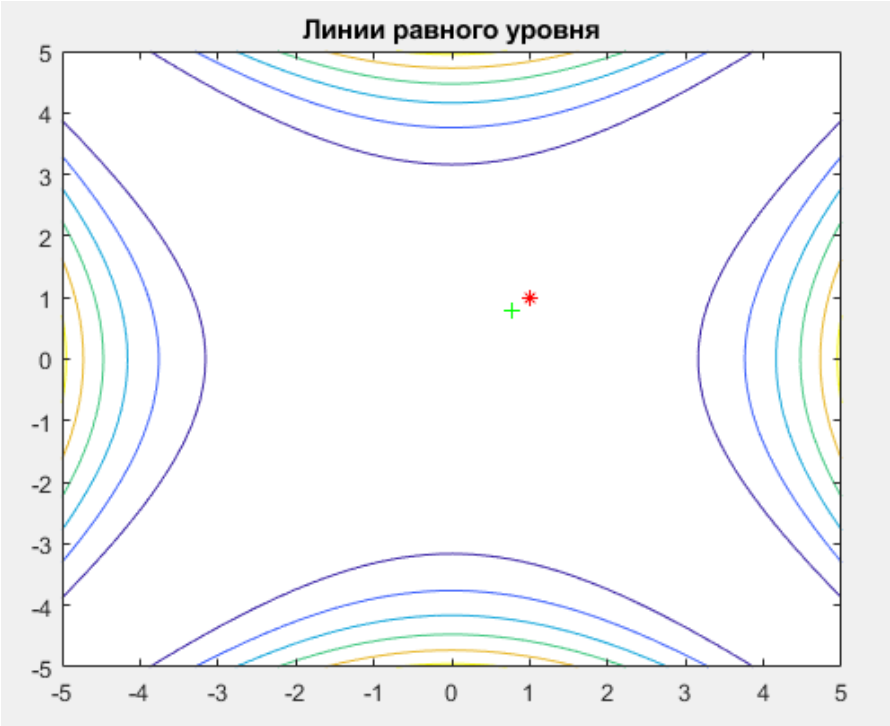

Рисунок 4 – Положение найденного оптимума x1 = 0.6064 и x2 = 0.6204

Функция Гриванка:

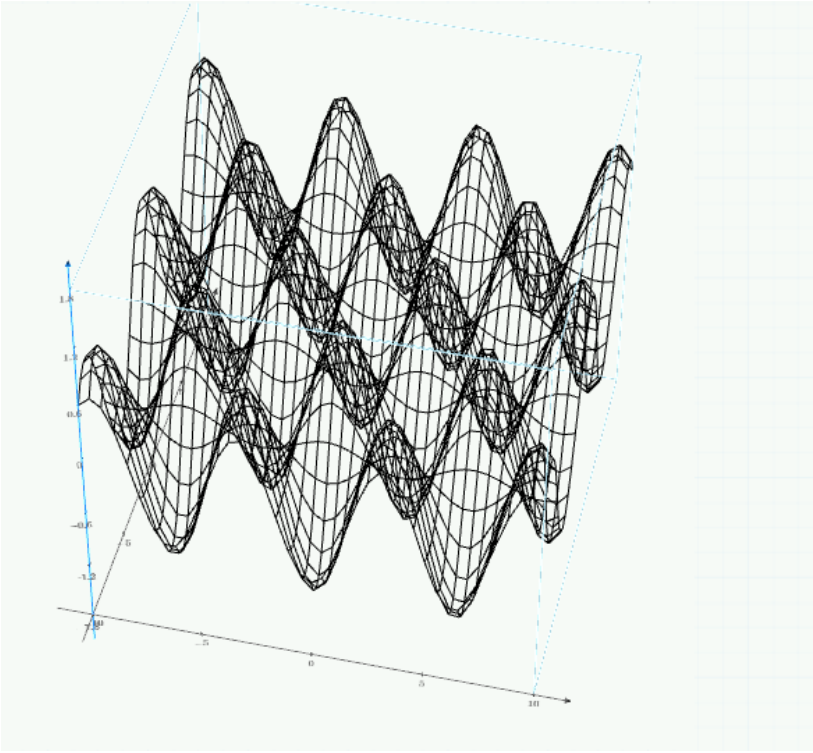

Рисунок 5 - Поверхность функции Гриванка

Рисунок 6 – График изменения популяции в среднем

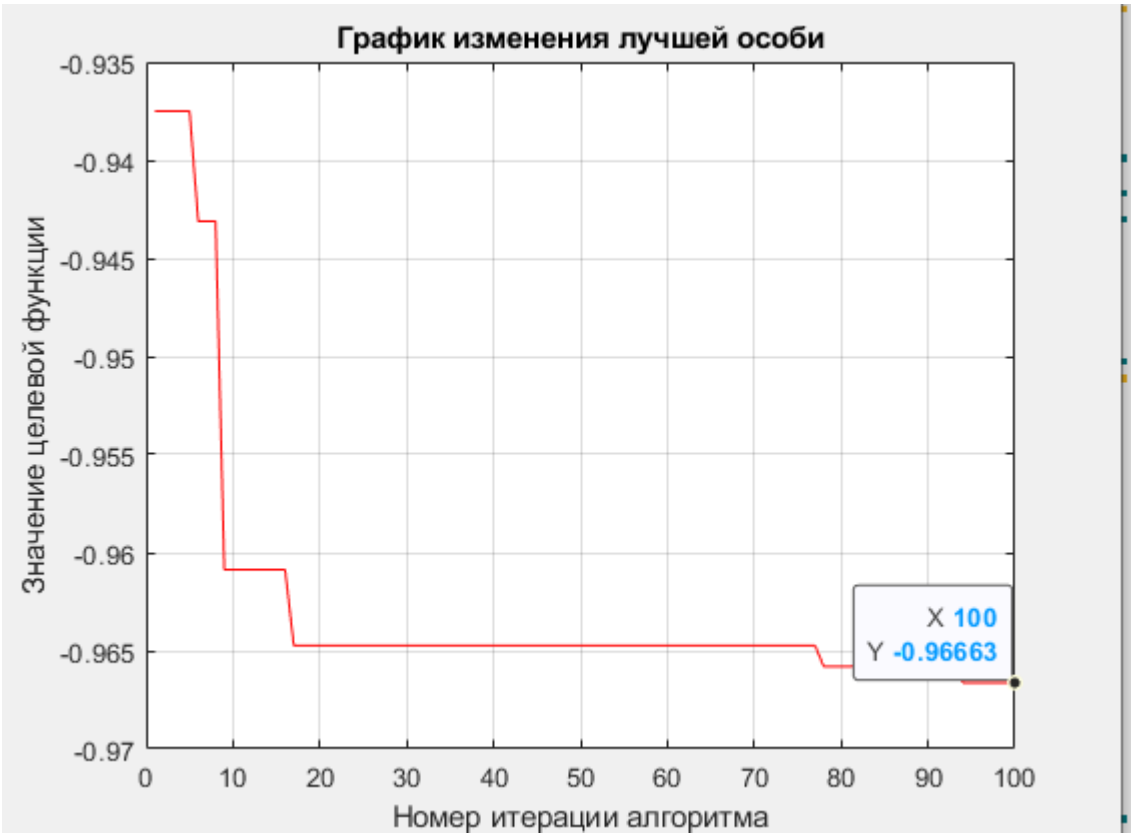

Рисунок 7 - График изменения популяции лучшей особи

Рисунок 3 – Исследование ГА на функции Гриванка а) поиск оптимума, б) поверхность целевой функции

Поиск оптимума функции функции Химмельблау:

Функция является многоэкстремальной, у нее 4 локальных минимума.

$$f(3,2)=0,$$
$$f(-2{,}805118\ldots,3{,}131312\ldots)=0,$$
$$f(-3{,}779310\ldots,-3{,}283186\ldots)=0,$$
$$f(3{,}584428\ldots,-1{,}848126\ldots)=0.$$

Алгоритм случайным образом определяет один из локальных минимумов.

Первый случай:

Рисунок 8 – График изменения популяции в среднем

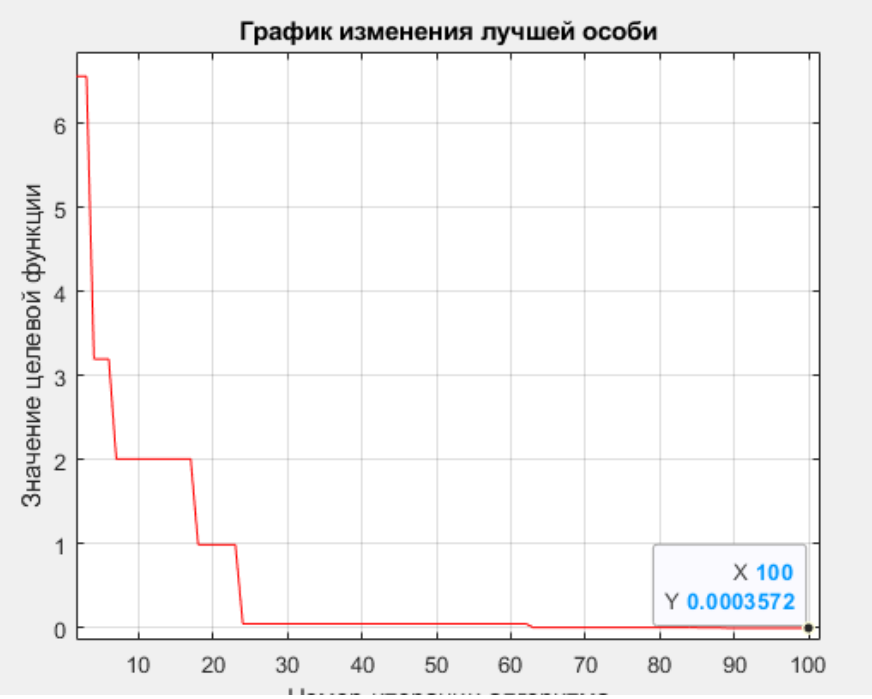

Рисунок 9 – График изменения популяции лучшей особи

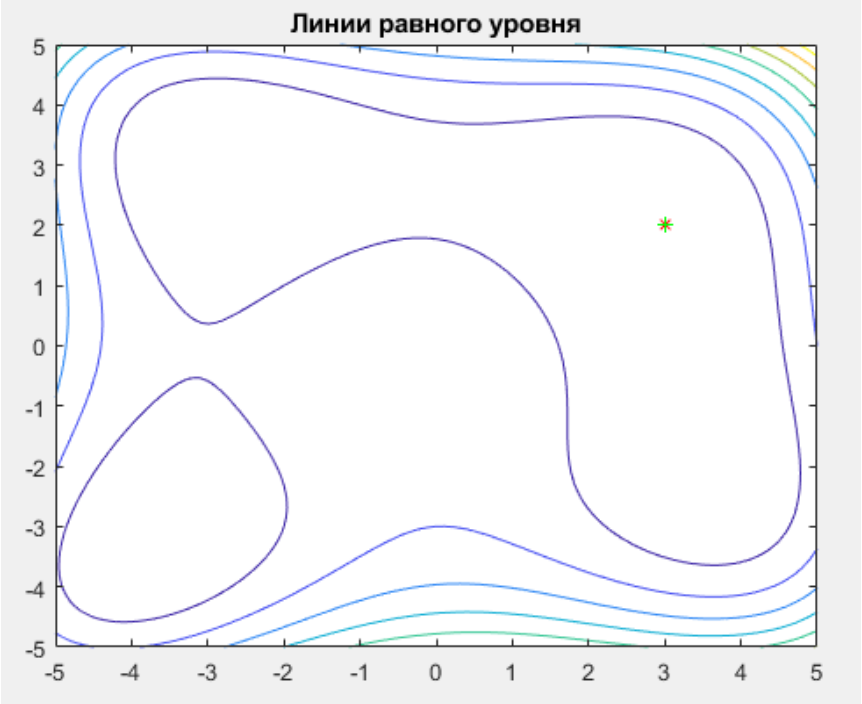

Рисунок 10 – Положение найденного оптимума x1 = - 2.9968 и x2 = 2.0035

Второй случай:

Рисунок 11 – График изменения популяции в среднем

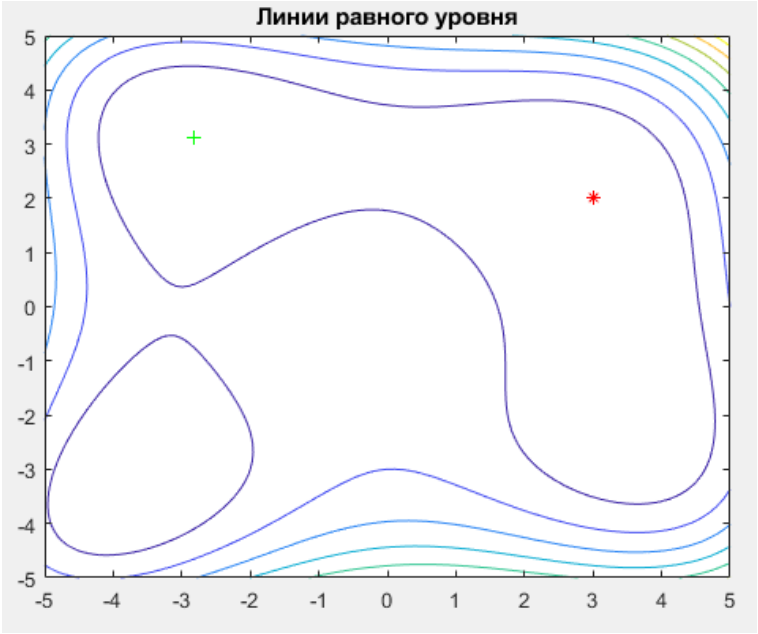

Рисунок 12 – Положение найденного оптимума x1 = -2.8117 и x2 = 3.1312

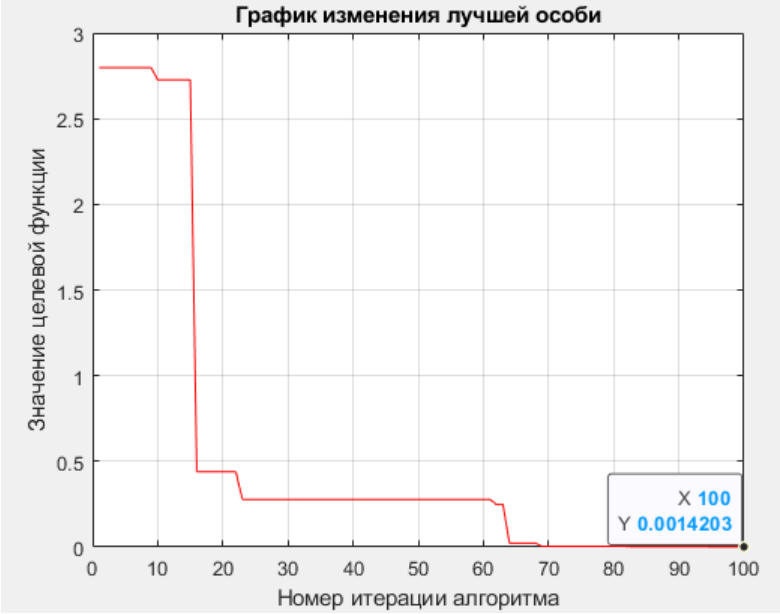

Рисунок 13 – График изменения популяции лучшей особи

Третий случай:

Рисунок 14 – График изменения популяции в среднем

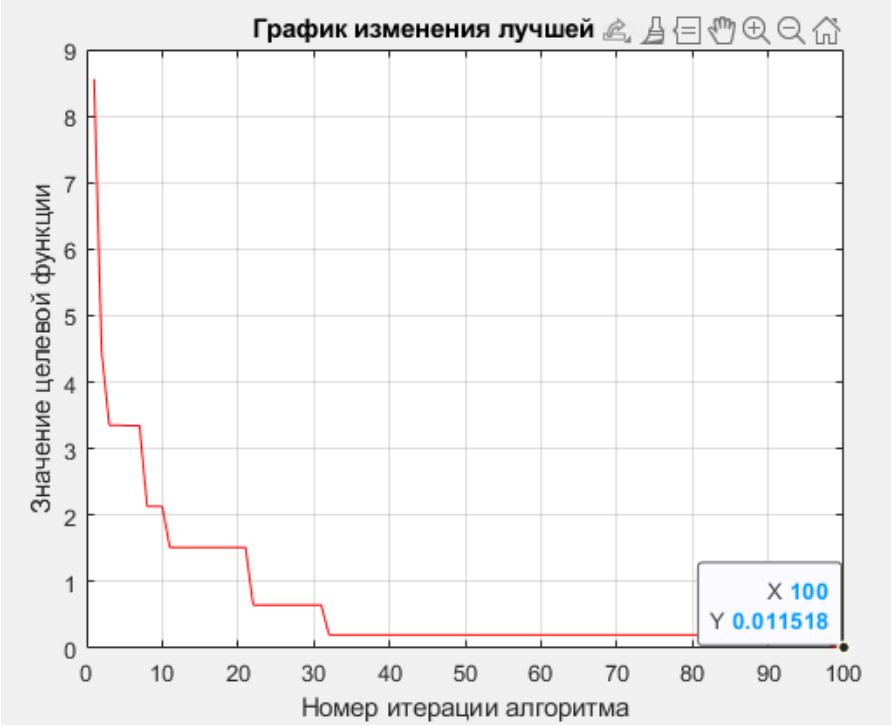

Рисунок 15 – График изменения популяции лучшей особи

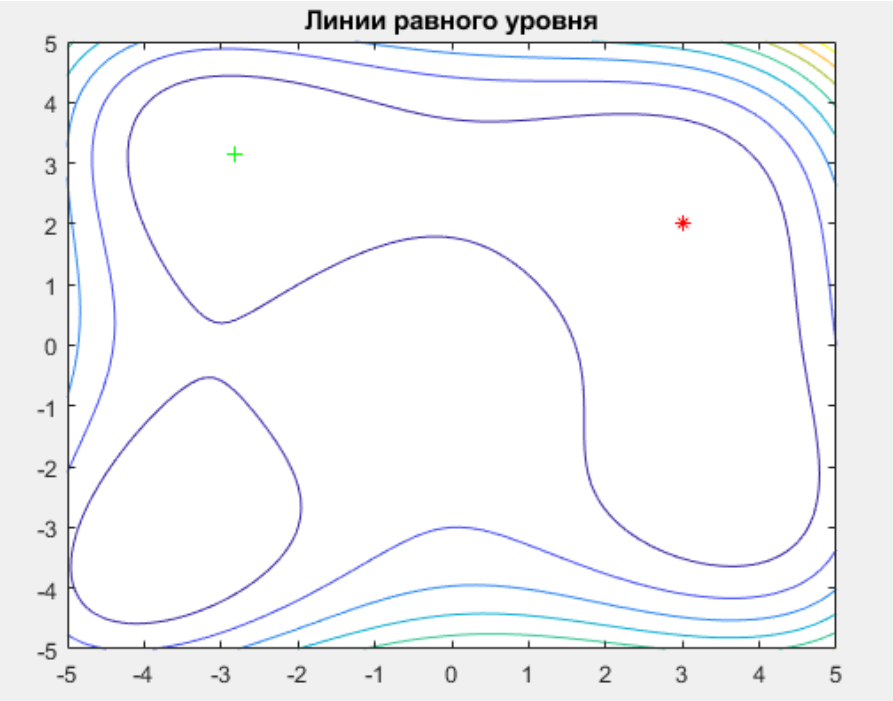

Рисунок 16 – Положение найденного оптимума x1 = -2.8236и x2 = 3.1344

Вывод: был реализован непрерывный генетический алгоритм в *MATLAB* и протестирован на многоэкстремальных функциях. Наилучшим образом реализованный ГА показал себя на функции Химмельблау.