БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

ГиЭВ

Лабораторная работа №1

Альромхин Джорж

гр.858301

Минск 2019

**Лабораторная работа №1**

Поиск экстремума функции одной переменной с помощью генетических алгоритмов с бинарным представлением особей

**Цель задания**

Знакомство с основными понятиями теории генетических алгоритмов. Изучение и реализация классического простого генетического алгоритма с ис- пользованием представления решений в форме бинарных строк. Классические генетические операторы, кроссинговер и мутация для бинарных особей. Постановка задачи Дана функция одной переменной y = f(t). Найти экстремум данной функ- ции на отрезке t ε [t0,t1] с помощью генетического алгоритма с бинарным пред- ставлением особей. Иллюстрировать графически динамику поиска экстремума. Программную реализацию выполнить в системе Matlab в виде программного m- модуля.

**Вариант 6**. Найти минимум функции f(t)=(0.5t-1.4)cos(0.5πt+1.1), t [-9,9].

Fitness Function

function y = FitnessFnc(x)

y =(0.5\*x(1) -1.4) \* cos(0.5\*pi\*x(1) + 1.1);

end

**Function plot**

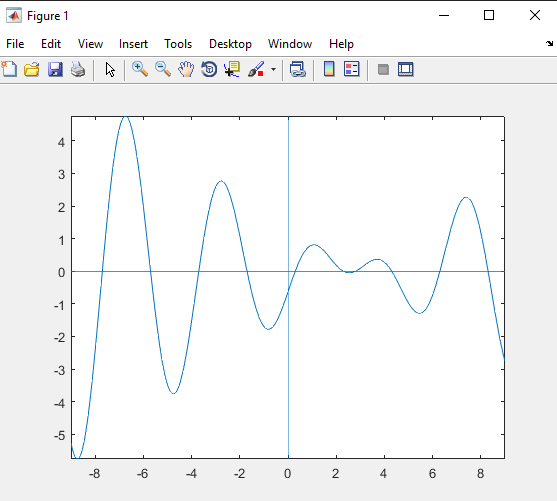
fplot(@(x) (0.5.\*x - 1.4).\*cos(0.5.\*pi.\*x + 1.1), [-9 9]);

xL = xlim;

yL = ylim;

line([0 0], yL); %x-axis

line(xL, [0 0]); %y-axis



Реализация функции ga

parabola = @FitnessFnc;

options = optimoptions('ga','PlotFcn','gaplotbestf');

%options

options.InitialPopulationRange = [-9 ; 9];

options.PopulationSize = str2double(get(handles.populationSize,'String'));

**Реализация функции кроссинговера**

options.CrossoverFraction = str2double(get(handles.crossover,'String'));

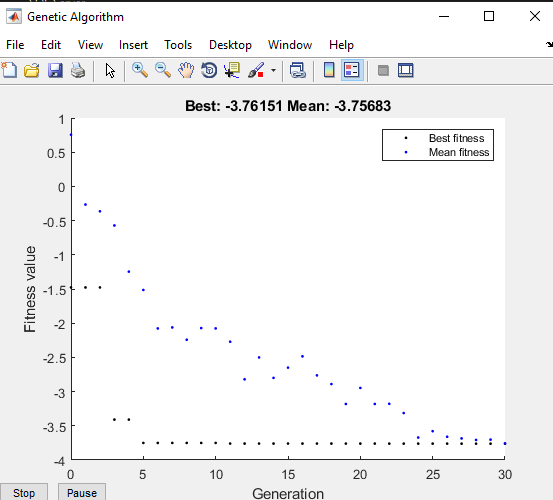
**Реализация функции мутации**

options.MutationFcn ={@mutationgaussian,str2double(get(handles.mutation,'String'))};

options.MaxGenerations = str2double(get(handles.maxGen,'String'));

**Зададим основные параметры алгоритма**

[x,fval,exitFlag,Output] = ga(parabola,str2double(get(handles.numVar,'String')),options);



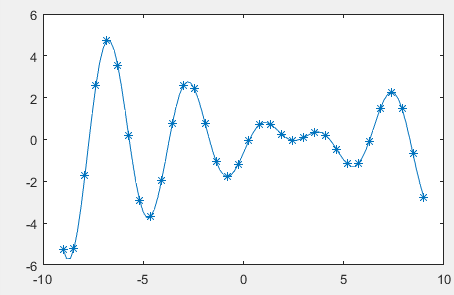
fval = -3.7602

axes(handles.axes1);

x = linspace(-9, 9);

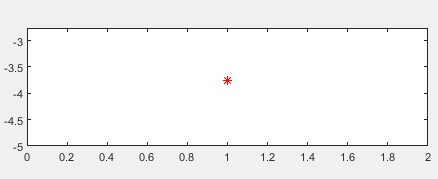
y = (0.5.\*x - 1.4).\*cos(0.5.\*pi.\*x + 1.1);

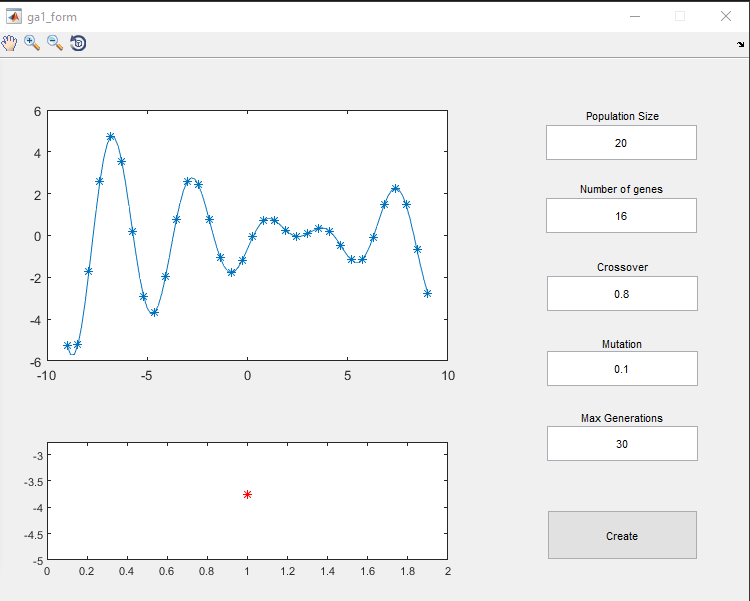
plot(x,y,'-\*','MarkerIndices',1:3:length(y))



axes(handles.axes2);

plot(fval,'r\*');





**Вывод**

В данной работе реализован алгоритм используя генетический алгоритм. Чтобы создать начальную популяцию, используйте методдробовик, собственнический выбор родителей, одиночн-точность crossingover и единственная мутация. Метод ломки используется для представления реальных решений. исходное решение разбивается на сегменты.