

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по учебной практике
Тема: Генетические алгоритмы и PSA

Студент гр. 2382

Федоров М.В.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2023

Цель работы.

Задание.

Задача о выделении области:

Дано N точек в двумерном евклидовом пространстве. Для каждой i -й точки задана метка: 0 или 1. Необходимо найти координаты левого нижнего угла, высоту и ширину прямоугольника, чтобы полученный прямоугольник содержал как можно больше точек с меткой 1, и как можно меньше точек с меткой 0.

Основные теоретические положения.

Задача ГА - найти оптимальное решение некоторой задачи, путем развития популяции потенциальных решений называемых индивидуумами. Решения итеративно оцениваются и используются для создания следующего поколения.

Генотип - описание одного индивидуума, набор генов сгруппированных в хромосому. При скрещивании хромосома содержит гены своих родителей.

Популяция - множество индивидуумов, то есть потенциальных решений, которые хранит генетический алгоритм. Популяция всегда отображает текущее поколение.

Функция приспособленности/целевая функция - функция, которую необходимо оптимизировать в рамках решаемой задачи. Является функцией от индивидуума, и показывает качество решения представленным хромосомой. На каждой итерации ГА рассчитывает приспособленность индивидуума для формирования нового поколения.

Отбор - формирование множества индивидуумов, которые будут использоваться для формирования следующего поколения. Отбор основывается на приспособленности индивидуума, и чем он лучше, тем больше вероятность его отобрать. Причем хромосомы дающие низкое значение приспособленности не исключают возможность отбора. Таким образом приспособленность популяции увеличивается.

Выполнение работы.

Для представления точек будет использоваться класс `Pointer`. Это класс содержит три поля, которые соответствуют характеристикам точки: координаты и метка. Так же реализованы сетеры для каждого поля.

Класс прямоугольника `Rectangle` задаётся двумя точками: левый верхний угол, правый нижний. Так же реализованы сетеры для каждого поля.

Целевая функция будет подсчитывать количество точек внутри прямоугольника. И возвращать кортеж значений: $(\text{num_point1} * n - \text{num_point0} * m, s)$ — где num_point1 , num_point0 , это количество точек с определёнными марками; n , m , некоторые числа(определим позже); s — площадь прямоугольника(этот параметр нужен, чтоб новый индивид не увеличивался в размерах).

Метод отбора не был выбран, поскольку хочется сравнить разные методы и найти эффективный. Но предположительно будет использоваться отбор усечением.

В качестве функции скрещивания будет использоваться одноточечное скрещивание. Новые прямоугольники будут получаться путём обмена точками, которые их задают.

В качестве мутации будет выбрана функция, которая увеличивает размер прямоугольника на n (позже определим точное значение).

Графический интерфейс

Графический интерфейс был разработан на языке программирования Python с использованием библиотек *customtkinter*, *matplotlib*, *PIL*, *tkinter*.

Основная часть интерфейса реализована с помощью библиотеки *customtkinter*, с помощью библиотеки *matplotlib* реализовано пошаговое отображение покрывающих плоскость прямоугольников и множество точек на графике. Библиотека *PIL* использовался для обработки изображений, установленных на кнопках взаимодействия. С помощью *tkinter* реализован функционал открытия *.csv файла, для считывания данных.

Примеры работы GUI представлены на рис. 1 - рис. 4.



Рисунок 1 — Установка начальных параметров

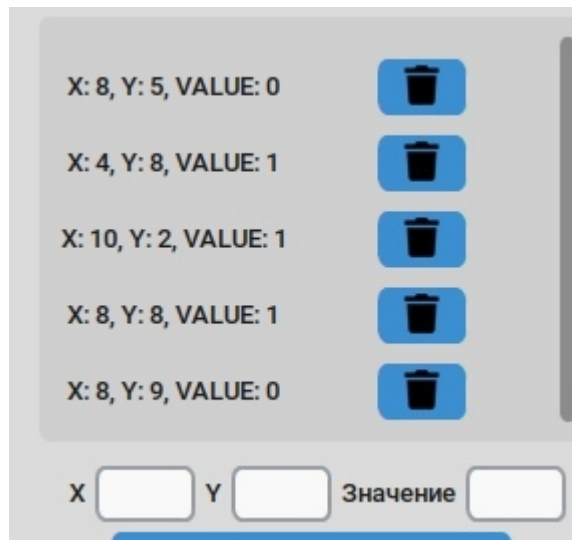


Рисунок 2 — Множество точек

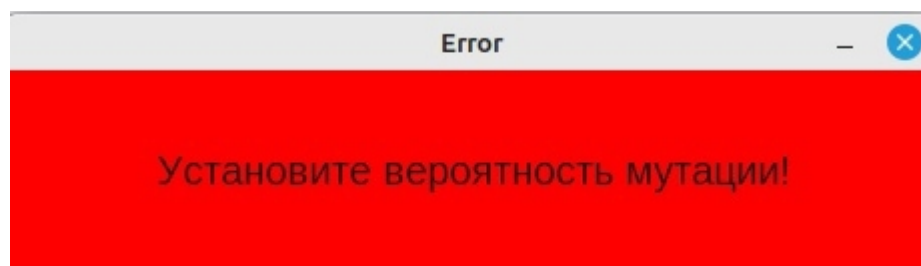


Рисунок 3 — Ошибка при запуске

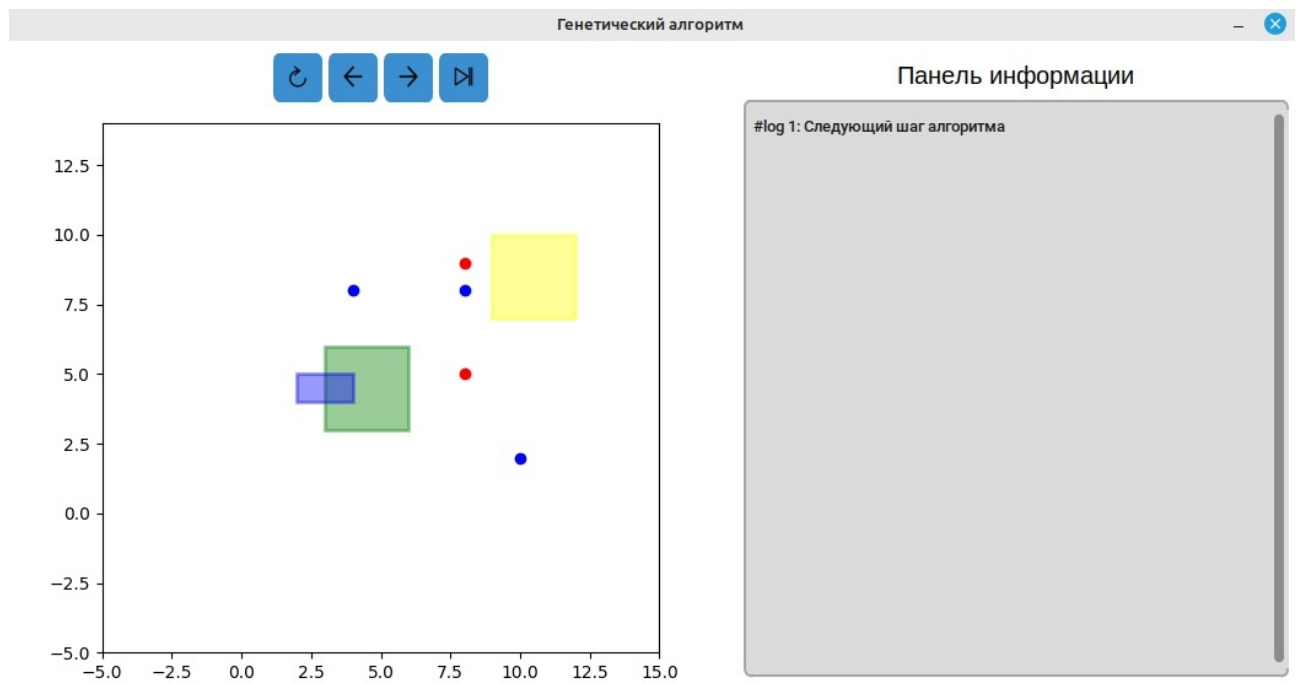


Рисунок 4 — Отображение работы алгоритма

Выводы.

ПРИЛОЖЕНИЕ