

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

ОТЧЕТ
по учебной практике
Тема: Генетические алгоритмы

Студент гр. 3343

Жучков О.Д.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2025

Цель работы

Изучение принципов генетических алгоритмов и их применения для решения оптимизационных задач, программная реализация генетического алгоритма для решения задачи и пользовательского интерфейса для него.

Задание

Вариант 12. Задача прямоугольного раскроя.

Дана полубесконечная полоса шириной N , необходимо разместить M прямоугольников, размеры i -го прямоугольника задаются пользователем как ширина w_i и длина l_i . Разместить прямоугольники необходимо так, чтобы потребовался участок полосы минимальной длины..

Требования к решению:

Программа должна иметь GUI

Должна быть возможность задать данные из разных источников по выбору пользователя: из файла, ввод через GUI, случайная генерация

При реализации алгоритмов нельзя использовать библиотеки решающие задачу напрямую. Генетический алгоритм должен быть реализован вручную. Использовать библиотеки для графического интерфейса/загрузки данных – можно.

Пользователь должен иметь возможность задавать параметры генетического алгоритма через GUI.

В приложении должна быть реализована пошаговая демонстрация работы генетического алгоритма. Т.е. должна быть кнопка «следующий шаг» и кнопка «выполнить до конца» (чтобы перейти к конечным результатам). На каждом шаге алгоритма должна быть показано графическое представление наилучшего решения, стоимость этого решения, средняя стоимость поколения.

В GUI должно быть поле с графиком изменения приспособленности лучший и средней в зависимости от поколения. График должен строиться по ходу выполнения генетического алгоритма.

После завершения выполнения генетического алгоритма, программа не должна закрываться, а должна быть возможность выполнить его на этих же данных, но с другими параметрами алгоритма, либо загрузить новые данные.

Выполнение работы

Программа для реализации алгоритма написана на языке Python. Для графического пользовательского интерфейса и вывода графиков используются библиотеки `tkinter` и `matplotlib`.

В файле `interface.py` описан класс, создающий окно `tkinter` и обеспечивающий пользовательское взаимодействие с программой. Пользователь может: ввести данные для решаемой задачи, поменять параметры генетического алгоритма, произвести пошаговое выполнение алгоритма и проследить за промежуточными результатами.

Основной файл `main.py` отвечает за создание объектов пользовательского интерфейса и исполнителя генетического алгоритма и осуществляет взаимодействие между ними.

В файле `model.py` описан класс, производящий решение задачи генетическим алгоритмом. Генетическое представление решения – перестановки чисел от 0 до N , которые сопоставляются с набором N прямоугольников, которые необходимо разместить на полосе. Для декодирования данного представления написан алгоритм, работающий следующим образом: берется i -ый прямоугольник из перестановки и размещается в ближайшее к началу доступное место (не занятое предыдущими $i-1$ прямоугольниками) на полосе (позиции рассматриваются слева направо, снизу вверх). Таким образом высчитывается длина участка, требуемая для размещения. Данный подход к кодированию решения исключает возможность некорректного решения или заведомо неоптимального. Число приспособленности равно длине участка, генетический алгоритм решает задачу нахождения минимума функции приспособленности.

Первым шагом алгоритма является создание случайного набора хромосом, каждый следующий шаг – создание новой популяции на основе предыдущей. С помощью ранжированной выборки определяется набор родителей, которые впоследствии скрещиваются с определенной вероятностью, их потомки могут мутировать. Реализованы два варианта условия окончания: превышение

определенного числа количеством поколений или схождение графиков средней и минимальной приспособленности (разность между средним и минимальным ниже определенного порога) Пользователю доступно изменение параметров алгоритма: размер популяции, количество популяций, нижний порог разности среднего и минимума, размер выборки, вероятности скрещивания и мутации.

В окне интерфейса пользователю предоставляются график средней и минимальной приспособленности каждого поколения, изображение наилучшего решения. Есть возможность приостановить и возобновить работу алгоритма, загрузить новые данные или параметры и запустить алгоритм заново.

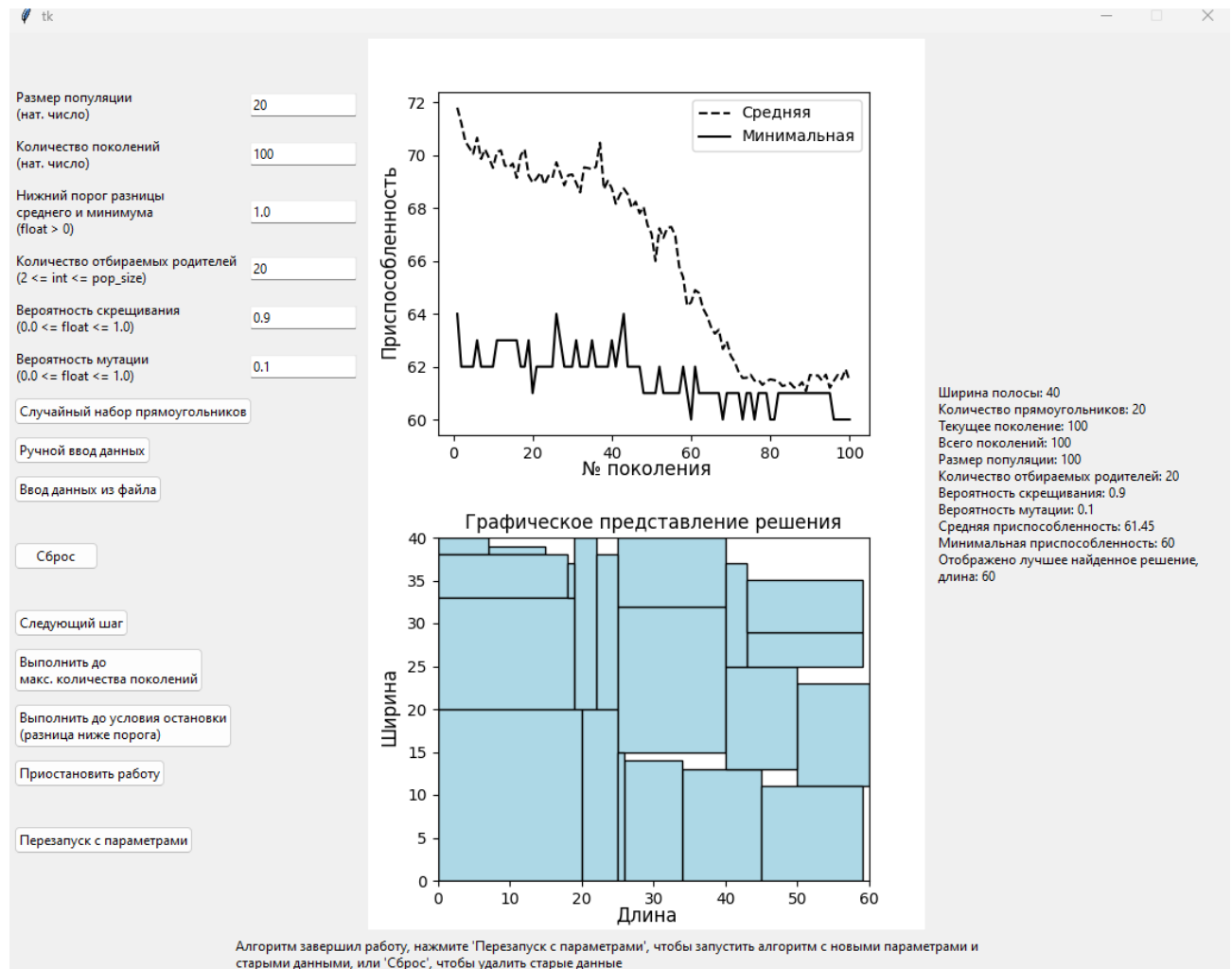


Рисунок 1 – Пример работы алгоритма