

**МИНОБРНАУКИ РОССИИ**  
**Санкт-Петербургский государственный**  
**электротехнический университет**  
**«ЛЭТИ» им. В.И. Ульянова (Ленина)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по учебной практике**  
**Тема: Генетические алгоритмы**

Студент гр. 3384

Питеров А. В.

Преподаватель

Жангиров Т. Р.

Санкт-Петербург

2025

## **Цели учебной практики**

- Ознакомиться с генетическими алгоритмами на примере решения оптимизационной задачи.
- Разработать полноценное приложение с GUI для работы над задачей, экспериментов с генетическими алгоритмами.

## **Задача (Вариант 16)**

Задано  $N$  фигур тетрамино (фигуры из игры тетрис) различных форм. Необходимо разместить их на поле шириной  $L$ , так, чтобы количество рядов, занимаемых фигурами, было минимально.

## **Содержание**

Итерация №1.....	3
Цели.....	3
Постановка работы.....	3
Компоненты генетического алгоритма.....	3
Работа над GUI.....	5

## **ИТЕРАЦИЯ №1**

### **Цели итерации**

- Определить как реализовывать ГА.
- Разработать прототип GUI.
- Срок до 29 июня.

### **Постановка работы**

Отчитаны требования к итоговой программе.

Установлено, что задачами программы является: решение пользовательской задачи с помощью ГА (в текущей работе — одну задачу данную вариантом) с возможностью настройки ГА (выбор параметров, конструирование из готовых блоков), демонстрировать поиск решения (графиками и визуализацией решения).

Отчитана теория касающаяся тетриса. Задача уточняется тем, что: фигуры бывают 7-ми типов (O, I, T, Z, S, L, J); их можно располагать на поле и вращать на  $90^\circ$ , они должны быть «прижаты к нижней части» поля.

Поле прямоугольное имеет один установленный параметр — ширину, длину предложено оценить, чтобы не работать с бесконечностью.

Целевая функция — количество занимаемых фигурами рядов для конкретного решения, может быть вычислена алгоритмически обходом поля.

Отчитана предложенная методичка.

## Компоненты генетического алгоритма

Так как задача состоит из фигур и поля, решение задачи можно представить в двух тривиальных видах, в табл. 1 приведено их сравнение.

Таблица 1 — Сравнение представлений

Представление 1	Представление 2
<b>Суть</b>	
Назначение каждой фигуре своей позиции на поле и ориентации.	Назначение каждой клетке поля принадлежности какой-либо фигуре.
<b>Параметры для кодировки</b>	
Позиция на поле и ориентация (4 поворота) — целые числа.	Принадлежность (каждой фигуре выдать свой номер) — целое число.
<b>Представление кодировки, затраты на память</b>	
Каждой фигуре сопоставить пару параметров и объединить в строку по некоторому принципу (последовательно, случайно). Память = количество фигур * вес параметров	Для каждой ячейки хранить параметр. Память = количество ячеек в поле * вес параметра
<b>Кодировка данных параметров влечёт наличие некорректных решений — проблемы кодировок</b>	
Выход фигуры за границы поля. Наложение фигур друг на друга.	Образование несуществующего типа фигуры. Фигура несвязна (разделена на части). Неправильное количество фигур.
<b>Как реагировать на некорректное решение?</b>	
Предполагается, что подавляющее количество решений будет некорректным в обоих представлениях. Поэтому отбрасывание приведёт к случайному поиску, исправление равноценно разработке переборных алгоритмов, а другие кодировки если и есть, то	

нетривиальны. Выбран подход со штрафами. Соотношения между штрафами — есть гиперпараметр, определяющий траекторию работы ГА.	
<b>Как проверить корректность решения?</b>	
Построение решения с фиксацией выхода за пределы поля и наложения.	Представлять исследуемую фигуру графом. Использовать обходы графов, изоморфизм, подсчитывать корректные фигуры.

Первый подход технически легче, поэтому будет реализован в первую очередь. Ожидается долгая сходимость, поэтому выбрано несколько операторов ГА для экспериментов, см. табл. 2.

Таблица 2 — Характеристики и операторы ГА

Популяция	Фиксированной длины с повторениями
<b>Оператор</b>	<b>Выбор (В порядке реализации)</b>
Отбор родителей	Панмиксия (случайно) Ин(аут)бридинг (метрика - расстояние Хэмминга)
Скращивание	Дискретная рекомбинация k-точечный кроссинговер
Мутация	Мутация инверсией (выборки, участков) Мутация обменом (генов, участков)
Отбор в новую популяцию	Отбор усечением (с/без элитаризма)

Как мера сохранения решений: банк на основе очереди с приоритетом фиксированной длины, использующийся при «плохих» преобразованиях популяции.

## Работа над GUI

Выделены следующие подходы к модификации поведения ГА: изменение исходного кода программы; выбор готовых решений для операторов ГА (встроенных в код программы, подключаемых извне — плагины); описание операторов как скриптов; конструирование потока исполнения ГА в визуальной форме. Решено реализовать второй подход как доступный.

При разработке интерфейса положено, что все касающиеся настроек должно быть убрано в отдельные меню-окна, а основное окно посвящено исключительно работе алгоритма. Так продуман интерфейс, перечисление всех окон и требований к ним см. табл. 3.

Таблица 3 — элементы GUI

Элемент	Описание
<b>Основное окно</b>	Содержит в себе ниже следующие панели
Панель управления	Содержит в себе кнопки для перехода к настройкам входящих данных и ГА, секции для управления ГА (следующий шаг, переход/пауза), для управления популяцией (сохранить/загрузить, сбросить), для управления графиками (сброс, сохранение).
Панель процесса	Содержит две вкладки: область просмотра и таблица популяции. Таблица агрегирует всю информацию рассчитываемую программой по каждому решению и позволяет вывести конкретное в область просмотра, которая позволяет увидеть его в графическом, кодированном и декодированном виде.
Панель информации	Содержит в себе секцию с требуемой информацией: номер популяции, средние и лучшие значения приспособленности и штрафа; секцию с графиками

	приспособленности и штрафов за весь процесс работы.
<b>Всплывающие окно настройки входных данных</b>	Вызывается при нажатии кнопки настройки данных на панели управления: позволяет настроить источники данных (ручной, файл и случайно) и выбрать конкретный для текущей работы.
<b>Всплывающие окно настройки ГА</b>	Вызывается при нажатии кнопки настройки ГА на панели управления: позволяет настроить и выбрать ГА для текущей работы. Настройка разбита по вкладкам, в соответствие с операторами ГА, в которых можно выбрать оператор из реализованных в коде программы.
<b>Всплывающие окно перехода</b>	Вызывается при нажатии кнопки переход на панели управления: позволяет выбрать ограничение в количестве популяций как критерий остановки процесса и запустить его.

Определены глобальные данные: выбранные для текущей работы данные и ГА (настройка и внутреннее состояние), текущая популяция, накопленная информация (построение графиков).

Определены связанные с ними потоки данных, что сведено в табл. 4.

Таблица 4 — Потоки данных

<b>Действие</b>	<b>Источник</b>	<b>Формат</b>
<b>Входные данные</b>		
Загрузить	Пользователь	Текстовый
<b>ГА (Настройка)</b>		
Загрузить/Сохранить	Пользователь/Программа	JSON
<b>Популяция (Решение)</b>		
Загрузить/Сохранить	Пользователь/Программа	Бинарный

Данные для графиков		
Сохранение	Пользователь	CSV

Определён порядок работы с программой, см. на рис. 1.

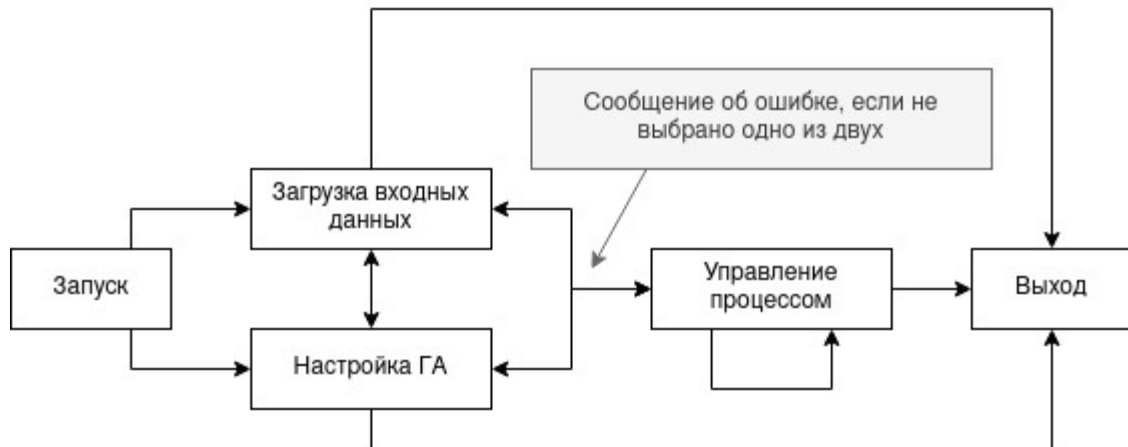


Рисунок 1 - порядок работы с программой

Выделены следующие способы реализации GUI:

- С использованием библиотеки на основном языке (C++)
- Связка со скриптовым языком (Python)
- web-интерфейс

Выбран первый подход, для чего сделан неглубокий обзор средств, в который попали, в скобках — причина отказа: Qt, wxWidgets, GTK (cairo для отрисовки), FLTK, ImGui (необходимость в GPU) и некоторые другие. Из них для данной работы выбран Qt, поскольку он де-факто стандарт и обладает готовыми средствами для отрисовки графиков.