

Университет ИТМО

Институт прикладных компьютерных наук  
Глубокое обучение и генеративный искусственный интеллект

ОТЧЕТ ПО 5-Й ЛАБОРАТОРНОЙ РАБОТЕ  
курса  
*«Эволюционные вычисления»*

ПРОЕКТИРОВАНИЕ ЭВОЛЮЦИОННОГО АЛГОРИТМА ДЛЯ ЗАДАЧИ  
РАССТАНОВКИ ФЕРЗЕЙ

Студент:  
*Группа № М4130*

*Батурина Ксения Александровна*

Санкт-Петербург 2024

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 РЕАЛИЗАЦИЯ И РЕЗУЛЬТАТЫ .....	4
1.1 Выбор фреймворка и эволюционного алгоритма .....	4
1.2 Анализ задачи и выделение критерия оптимизации .....	4
1.3 Представление решения .....	4
1.4 Условия терминции и характеристики для измерения эффективности .....	5
1.5 Настройка параметров и результаты экспериментов .....	5
2 ВОПРОСЫ .....	6
2.1 Является ли задача оптимизационной или ограниченной? .....	6
2.2 Как растет сложность задачи при увеличении размерности? ...	6

## ВВЕДЕНИЕ

Код доступен в репозитории на GitHub: [https://github.com/xeniabaturina/ITMO\\_EVOL/tree/main/lab5](https://github.com/xeniabaturina/ITMO_EVOL/tree/main/lab5).

Цель работы:

Освоить весь цикл разработки эволюционных алгоритмов, начиная с анализа проблемы и проектирования, заканчивая настройкой параметров и анализом эффективности.

Задачи работы:

1. Выбрать фреймворк (можно продолжить в Watchmaker).
2. Выбрать эволюционный алгоритм (ГА или другой).
3. Проанализировать задачу, выделить критерий для оптимизации или ограничения.
4. Спроектировать решение поставленной задачи, вид решения, способы вариаций и обработки возможных ограничений.
5. Сформировать условия терминации или сходимости к алгоритму.
6. Установить характеристики для измерения эффективности алгоритма.
7. Настроить параметры алгоритма.
8. Провести серии запусков при разных значениях  $N$  и провести анализ эффективности.
9. Ответить на вопросы:
  - Является ли задача оптимизационной или ограниченной?
  - Как растет сложность задачи при увеличении размерности?

# 1 РЕАЛИЗАЦИЯ И РЕЗУЛЬТАТЫ

## 1.1 Выбор фреймворка и эволюционного алгоритма

Для решения задачи были выбраны фреймворк Watchmaker и генетический алгоритм.

## 1.2 Анализ задачи и выделение критерия оптимизации

Основная задача состоит в такой расстановке  $N$  ферзей на доске, чтобы исключить взаимные атаки по горизонтали, вертикали и диагонали. Критерием оптимизации была выбрана минимизация количества конфликтов между ферзями.

## 1.3 Представление решения

Решение задачи — массив, где каждый элемент соответствует одному ряду шахматной доски, а значение элемента — номеру столбца, в котором находится ферзь в данном ряду. Таким образом, каждая конфигурация доски может быть однозначно описана этим массивом, это обеспечивает простое и эффективное представление решения.

Оператор кроссовера (`QueenCrossover`) ответственен за создание новых потомков из двух родителей (решений). Он случайным образом выбирает точку и обменивает две части родительских решений для формирования двух потомков.

Оператор мутации (`QueenMutation`) выполняет случайные изменения в решении, представляя случайное перемещение ферзя в другой столбец.

Фитнес-функция (`QueenFitnessFunction`) оценивает качество расстановки ферзей. Эта функция подсчитывает количество конфликтов, где конфликтом считается совпадение по горизонтали, вертикали или диагонали.

## 1.4 Условия терминции и характеристики для измерения эффективности

Процесс эволюции завершается после заданного числа поколений.

Основной характеристикой эффективности является количество конфликтов в лучшем найденном решении. Также важно учитывать количество итераций, потребовавшихся для сходимости.

## 1.5 Настройка параметров и результаты экспериментов

В соответствии с заданием была проведена серия запусков для решения задачи с размерностями 4, 8, 16, 32 и 64. В таблице ниже приведены размер доски, размер популяции, количество итераций алгоритма для всех запусков и количество конфликтов. Были подобраны оптимальные параметры алгоритма, такие как размер популяции и количество поколений, с учетом требований экспериментов.

Таблица 1 — Результаты производительности алгоритма.

N	Размер популяции	Количество итераций	Количество конфликтов
4	5	31,9	0,2
8	10	376,4	0,8
16	10	1210,7	1,9
32	5	2819,0	3,6
64	2	7682,8	6,4

## **2 ВОПРОСЫ**

### **2.1 Является ли задача оптимизационной или ограниченной?**

Рассматриваемая задача является оптимизационной, поскольку стремится к минимизации числа конфликтов.

### **2.2 Как растет сложность задачи при увеличении размерности?**

Сложность задачи увеличивается экспоненциально с ростом размерности доски, что подчеркивает значимость эволюционных методов для решения подобных задач.