**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«Национальный исследовательский университет ИТМО»**

**(Университет ИТМО)**

**Факультет Институт прикладных компьютерных наук**

**Образовательная программа 01.04.02**

Отчет

по лабораторной работе №5

по дисциплине **«Генетические алгоритмы»**

Выполнил: **студент группы M4130 Горбатовский А. В.**

Проверил: **Муратов С. Ю.**

Санкт-Петербург

2024

# Проектирование эволюционного алгоритма для задачи расстановки ферзей

## Цель работы.

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

## Ход работы.

### Был использован фреймворк watchmaker.

### В процессе выполнения лабораторной работы, целью которой было найти способ размещения n ферзей на шахматной доске размером n x n так, чтобы они не угрожали друг другу, особое внимание было уделено методу представления решений. Важным условием является избежание расстановок, при которых все ферзи находятся на одной линии, так как такие варианты не являются корректными. Учитывая это, для представления решений был выбран подход, основанный на генерации случайных перестановок числового массива от 0 до n, что исключает повторения в строках и столбцах. В этом контексте, основная задача заключается в проверке, не атакуют ли ферзи друг друга по диагоналям.

### Код, реализующий этот подход, создает объект, который генерирует случайную перестановку для заданной размерности доски. Это достигается путем создания списка чисел от 0 до n-1, который затем случайным образом перемешивается. Полученный массив чисел представляет собой одно из возможных решений задачи, где индекс массива соответствует номеру строки, а значение в этой позиции - номеру столбца, в котором должен стоять ферзь. Таким образом, каждое решение уникально и удовлетворяет заданным условиям задачи.

### Чтобы оценить эффективность предложенного решения расстановки ферзей на шахматной доске, применяется функция подсчета количества конфликтов, возникающих, когда два ферзя находятся на одной диагонали. Этот подход позволяет точно определить, насколько хорошо сгенерированное решение соответствует задаче: чем меньше таких пар ферзей, тем лучше решение. Реализация этой идеи осуществляется через функцию, которая анализирует каждую пару ферзей в решении на предмет возможных диагональных атак. Эффективность решения определяется суммарным количеством таких конфликтов: идеальное решение не имеет конфликтов, что означает отсутствие взаимных атак ферзей по диагоналям. Этот метод подчеркивает важность не только избегания пересечений в линиях и столбцах, но и необходимость контроля за диагональными угрозами.

### В разделе о мутации рассматривается процесс внесения случайных изменений в существующие решения для обеспечения разнообразия в популяции. Этот процесс заключается в выполнении случайных попарных обменов элементов в массиве решения. Количество таких обменов определяется заранее и может варьироваться. Основная цель мутации - создание новых вариантов решений без нарушения основной структуры перестановки, что позволяет избежать повторения уже существующих решений и способствует поиску оптимального расположения ферзей на доске.

### Кроссовер включает в себя выбор определенного участка из первого родителя и заполнение оставшихся позиций элементами из второго родителя, сохраняя при этом порядок этих элементов. Для реализации этого подхода определяются две случайные точки на доске, которые обозначают начало и конец подпоследовательности, которая будет скопирована из первого родителя. Этот выбранный участок копируется прямо в потомство без изменений. Затем мы заполняем оставшиеся позиции в потомстве, используя элементы из второго родителя, но при этом убеждаемся, что эти элементы располагаются в соответствии с их порядком во втором родителе. Важно отметить, что при заполнении избегается дублирования элементов, чтобы каждый ферзь оставался в уникальной позиции.

### В рамках лабораторной применяется генетический алгоритм, который характеризуется использованием RankSelection в качестве оператора селекции. Ключевые параметры алгоритма включают в себя размер популяции и количество мутаций. Алгоритм продолжает работу до тех пор, пока не наступит стагнация процесса на протяжении 1000 итераций, что служит критерием остановки.

### Результаты тестирование представлены в таблице 1.

Таблица 1 – Результаты экспериментов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Размер популяции | Кол-во мутаций | Кол-во генераций | Значение фитнес функции |
| 4 | 10 | 2 | 1.20 | 0 |
| 8 | 10 | 2 | 7.90 | 0 |
| 16 | 10 | 2 | 15.40 | 0.2 |
| 32 | 10 | 2 | 13.80 | 0.40 |
| 64 | 10 | 2 | 29.30 | 0.60 |
| 128 | 10 | 10 | 55.70 | 0.40 |

## Ответы на вопросы:

### Задача, не сосредоточена на оптимизации в традиционном понимании, поскольку главная цель не в улучшении показателя фитнес функции, а в использовании его в качестве средства для достижения точного решения. Это позволяет определить пригодность решения без неопределенности.

### Влияние размерности на сложность вычисления функции приспособленности пропорционально квадрату размерности (O(n^2)), что указывает на то, что с ростом размерности задачи увеличивается и сложность ее решения. Для эффективного поиска решения при больших размерностях требуется расширение популяции, что, в свою очередь, приводит к увеличению количества вычислений функции приспособленности.

Ссылка на репозиторий: <https://github.com/Myashka/ITMO_Gen_Lab_5>