

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

ОТЧЕТ
по учебной практике
Тема: Генетические алгоритмы. Задача о рюкзаке

Студент гр. 0303

Афанасьев Д.В.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2022

Цель работы.

Разработать и реализовать программу, решающую задачу о рюкзаке с использованием генетических алгоритмов (ГА), а также визуализирующую работу алгоритма.

Задание.

Задача о рюкзаке. Требуется уложить как можно большее число ценных вещей в рюкзак при условии, что вместимость рюкзака ограничена.

Входные данные:

- Размер рюкзака;
- Количество предметов, их веса и стоимость.

В качестве языков программирования можно выбрать: C++, C#, Python, Java, Kotlin. Все параметры ГА необходимо определить самостоятельно исходя из выбранного варианта. Разрешается брать один и тот же вариант разным бригадам, но при условии, что будут использоваться разные языки программирования.

Требования к программе:

- Наличие графического интерфейса (GUI);
- Возможность ввода данных через GUI или файла;
- Пошаговая визуализация работы ГА. С возможностью перейти к концу алгоритма;
- Настройка параметров ГА через GUI. Например, размер популяции;
- Одновременное отображение особей популяции с выделением лучшей особи;
- Визуализация кроссинговера и мутаций в популяции;
- Наличие текстовых логов с пояснениями к ГА;
- Программа после выполнения не должна сразу закрываться, а должна давать возможность провести ГА заново, либо ввести другие данные.

Описание представления данных

Для представления хромосомы используется булевый вектор, в котором каждый элемент — это предмет (ген), который можно положить в рюкзак. Если в хромосоме на i -ой позиции стоит 1, то это значит, что i -ый предмет положили

в рюкзак, если 0, то нет. Стоит отметить, что при такой реализации хромосомы, не каждая хромосома будет принадлежать допустимому решению, то есть вес всех предметов в хромосоме может быть больше, чем вместимость рюкзака.

Описание алгоритма

Перед началом работы алгоритма происходит генерации n хромосом (n задается пользователем), удовлетворяющих допустимому решению.

Ключевая часть алгоритма состоит из набора последовательных действий, описанных ниже, которые повторяются до тех пор, пока в последних полученных потомствах максимальная приспособленность не перестанет меняться (количество потомств может задаваться пользователем) или количество потомств не достигнет количества, ограниченного пользователем.

Действия:

1. Определение приспособленности каждой хромосомы. Определяется при помощи фитнес-функции, которая вычисляется по формуле $f(\vec{x}) = \sum_{i=1}^n c_i \cdot x_i$, где x_i — наличие i -го предмета в хромосоме, $x_i \in \{0,1\}$, а c_i — стоимость i -го предмета.
2. Турнирный отбор. При помощи отбора определяются особи для кроссинговера (у пользователя есть возможность определить количество особей участвующих в турнире). Отбор происходит по следующему правилу: отбирается t хромосом и из них выбирается хромосома с максимальной приспособленностью, после чего она добавляется в список для кроссинговера. Такая операция повторяется n раз.
3. Кроссинговер. После отбора особей происходит одноточечный кроссинговер между ними. Берутся две случайные особи, первая половина генов первой особи соединяется со второй половиной второй особи и наоборот. Таким образом, получаются две новые особи, которые будут входить в потомство.

4. После кроссинговера происходит проверка потомства на допустимость решения каждой особи. Если особь недопустима, то случайный не нулевой ее ген меняется на 0. Данная операция повторяется до тех пор, пока особь не станет допустимой.

5. Завершающим этапом является мутация потомства, которая с определенной вероятностью инвертирует один случайный ген в хромосоме. Вероятность мутации может задаваться пользователем.

После выполнения пятого действия получается итоговое потомство от текущей популяции.

Описание взаимодействия пользователя с программой

1) В начале пользователь попадает на главное меню (рис. 1), где ему предоставляется выбор способа ввода входных данных: из файла или при помощи GUI.

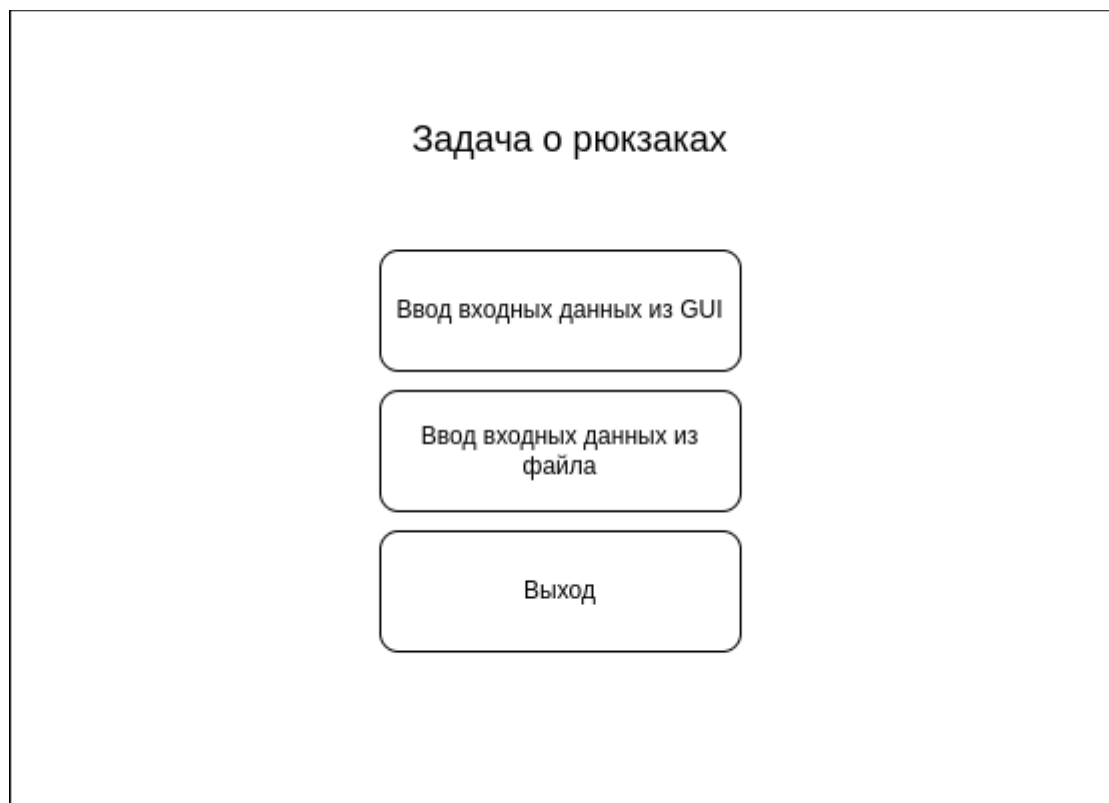


Рисунок 1 — Главное меню

2) Если пользователь выбрал ввод данных при помощи GUI, то перед ним откроется меню (рис. 2), где он сможет ввести вместимость рюкзака,

количество предметов, их вес и стоимость. Если он выбрал ввод входных данных из файла, то перед ним откроется меню (рис. 3), где он сможет ввести путь до нужного файла.

файл | о программе

Введите вместимость рюкзака:

Введите количество предметов:

Стоимость	Вес
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

равно количеству предметов
появится после нажатия
кнопки ввести

Рисунок 2 — Меню ввода данных при помощи GUI

файл | о программе

Введите путь до файла:

Рисунок 3 — Меню для указания пути до файла

3) После ввода входных данных пользователю предоставляется возможность выбрать параметры ГА (рис. 4). Список параметров: размер популяции, количество элементов для участия в турнирном отборе, вероятность мутации, условие для завершения работы алгоритма.

The image shows a graphical user interface for a Genetic Algorithm (GA) application. At the top, there are two tabs: "файл" (file) and "о программе" (about the program). Below the tabs, there are four input fields, each preceded by a label: "Введите размер популяции:" (Enter population size:), "Введите количество особей в отборе" (Enter number of individuals in selection), "Введите вероятность мутации:" (Enter mutation probability:), and "Введите seed для генерации:" (Enter seed for generation:). Each label is followed by a rectangular text input box. In the bottom right corner of the window, there is a rounded rectangular button with the text "перейти к алгоритму" (go to algorithm).

Рисунок 4 — Меню для указания параметров ГА

4) Когда указаны все параметры входных данных, происходит переход к окну выполнения алгоритма (рис. 5). У пользователя есть возможность перехода к следующему ходу или к итоговому решению (возможно будет возможность возвращаться к предыдущим шагам). Во время работы алгоритма пользователю также будут представлены текущие особи, их потомства, лучшие особи, а также информация о том, между какими особями происходит кроссинговер.

файл	о программе																																																																																
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><th colspan="5">Текущая популяция</th></tr> <tr><td style="width: 10%;">1</td><td style="width: 10%;">1</td><td style="width: 10%;">0</td><td style="width: 30%;"></td><td style="width: 10%;">0</td></tr> <tr><td colspan="5" style="height: 20px;"></td></tr> <tr><td colspan="5" style="height: 20px;"></td></tr> <tr><td>1</td><td>0</td><td>0</td><td></td><td>1</td></tr> <tr><td colspan="5" style="height: 40px;"></td></tr> <tr><td>1</td><td>1</td><td>0</td><td></td><td>1</td></tr> <tr><td colspan="5" style="height: 20px;"></td></tr> </table>	Текущая популяция					1	1	0		0											1	0	0		1						1	1	0		1						<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><th colspan="5">Потомство</th></tr> <tr><td style="width: 10%;">1</td><td style="width: 10%;">1</td><td style="width: 10%;">0</td><td style="width: 30%;"></td><td style="width: 10%;">0</td></tr> <tr><td colspan="5" style="height: 20px;"></td></tr> <tr><td colspan="5" style="height: 20px;"></td></tr> <tr><td colspan="5" style="height: 20px;"></td></tr> <tr><td colspan="5" style="height: 40px;"></td></tr> <tr><td colspan="5" style="height: 20px;"></td></tr> <tr><td colspan="5" style="height: 20px;"></td></tr> </table>	Потомство					1	1	0		0																														
Текущая популяция																																																																																	
1	1	0		0																																																																													
1	0	0		1																																																																													
1	1	0		1																																																																													
Потомство																																																																																	
1	1	0		0																																																																													
Место с основной информацией о произошедшем на текущем шаге																																																																																	
<div style="border: 1px solid black; border-radius: 10px; padding: 10px 40px; display: inline-block;">предыдущий шаг</div>	<div style="border: 1px solid black; border-radius: 10px; padding: 10px 40px; display: inline-block;">текущий шаг</div>	<div style="border: 1px solid black; border-radius: 10px; padding: 10px 40px; display: inline-block;">перейти к итоговму решению</div>																																																																															

Рисунок 5 — Окно работы ГА

5) После завершения алгоритма пользователь будет перенаправлен на первоначальное меню.

Основные классы, их поля и методы, для рализации ГА

- *class DataGA* — класс предназначенный для хранения основных данных, которые задает пользователь.

Поля:

- `std::vector<std::pair<unsigned int, unsigned int>>` items — массив предметов, которые можно положить в рюкзак. Каждый предмет задан парой, где первое число — стоимость, второе — вес;
- `unsigned int sizePopulation` — размер популяции;
- `unsigned int capacity` — объем рюкзака;
- `unsigned int seed` — число для генерации данных;
- `unsigned int numIndividualsSelection` — число особей участвующих в отборе;

- `float probabilityMutation` — вероятность того, что одна особь мутирует, задается число от 0 до 1.

Методы — для каждого поля существует, соответствующий метод `get` и `set`. Такой принцип был выбран для того, чтобы в класс, выполняющий ГА, не мог никак изменить входные данные, так как ему будет передан константный экземпляр класса `DataGA`.

- `class Chromosome` — класс, описывающий хромосому (особь).

Поля:

- `const DataGA& data` — набор параметров, введенных пользователем;
- `std::vector<bool> genes` — массив генов, где каждый ген — это предмет, который можно положить в рюкзак. 0 означает его отсутствие в рюкзаке, а 1 наличие.

Методы:

- `unsigned int fitnessFunction()` — фитнес-функция, показывает приспособленность особи;
- `void correction()` — метод для коррекции генов особи, так как особь может не принадлежать допустимому решению;
- `void mutation()` — метод, производящий мутацию особи с заданной вероятностью;
- `bool isAcceptable()` — метод проверяет принадлежит ли особь допустимому решению;
- `std::vector<bool> getGenes()` — метод, возвращающий массив генов особи.
- `class GA` — класс, описывающий основные действия работы ГА.
- Поля:
 - `const DataGA& data` — набор параметров, введенных пользователем;
 - `std::vector<Chromosome> population` — текущая популяция, на основе, которой будет получено потомство;
 - `std::vector<Chromosome> child` — потомство;

- `std::vector<int> inSelection` — особи, участвующие в кроссинговере;
- Методы:
 - `void selection()` — метод, производящий отбор особей для кроссинговера;
 - `void mutation()` — метод, производящий мутации потомство;
 - `void crossing()` — кроссинговер, между особями полученными при помощи метода *selection*;
 - `void transfer()` — метод для замены родителей текущим потомством;
 - `void pick()` — метод для отбора лучшей особи среди *t* случайных особей.

GUI

В ходе работы был разработан интерфейс, он представлен на рис. 6-10.

В первом окне (рис. 6) пользователю предоставляется возможность выбрать тип ввода данных.

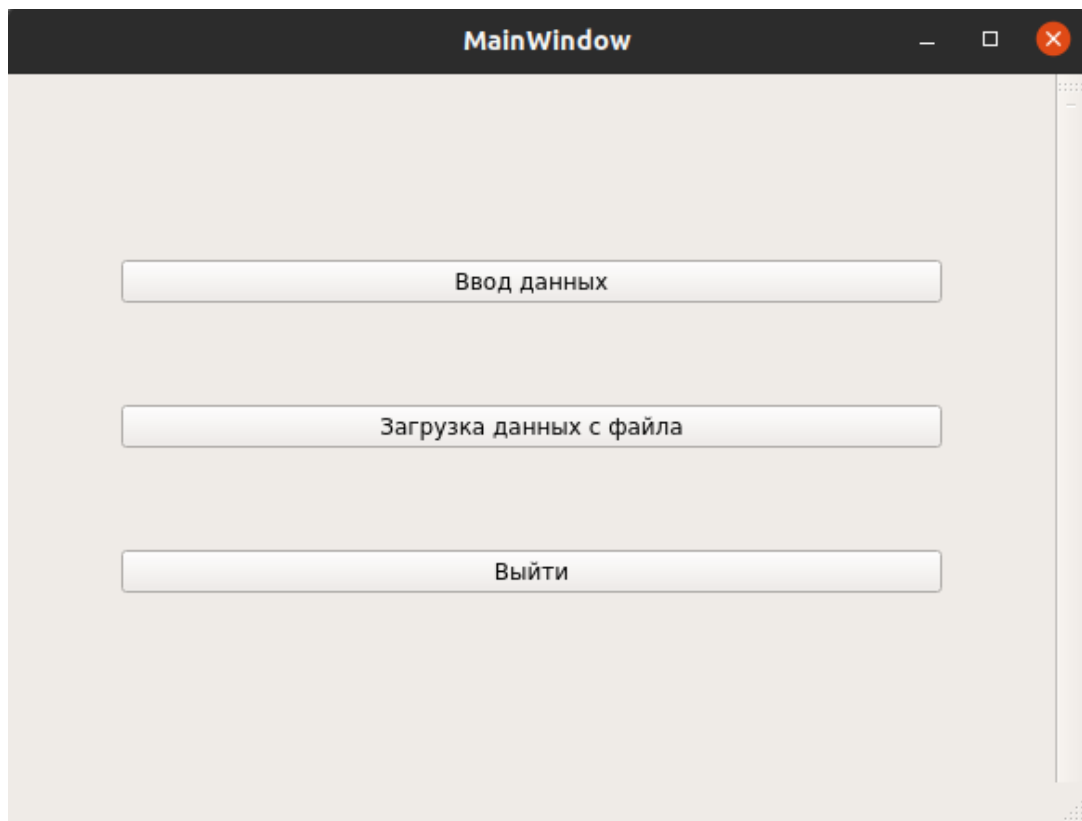


Рисунок 6 — Окно выбора типа ввода данных

На рис. 7 представлено окно, на котором пользователь вводит данные задачи. Если количество строк (вес и стоимость предмета), которые ввел пользователь, не равно тому количеству предметов, то он не будет переведен на окно ввода параметров ГА.

The screenshot shows a window titled 'MainWindow' with a standard Windows-style title bar (minimize, maximize, close buttons). The window has a tabbed interface with the first tab labeled 'Основные'. Below the tab bar, the text 'Ввод данных о задаче' is displayed. There are two input fields: 'Вместимость рюкзака' (Backpack Capacity) with a value of 7, and 'Количество предметов' (Number of items) with a value of 8. Below these fields, there is a text area with the instruction: 'Введите предметы, где каждый предмет записан в одной строке. В каждой строке первое число - стоимость, второе - вес.' (Enter items, where each item is recorded on one line. In each line, the first number is the cost, the second is the weight). The text area contains the following data:

2	1
3	2
4	4
3	5
4	3
6	7
2	2
1	10

A button labeled 'Ввод параметров ГА' (GA parameter input) is located at the bottom right of the window.

Рисунок 7 — Окно ввода данных при помощи GUI

На рис. 8 представлено окно, на котором пользователь вводит путь до файла, из которого будут считаны входные данные задачи. Если данные не будут корректны или файла не существует, то пользователь не будет перенаправлен на следующее окно.

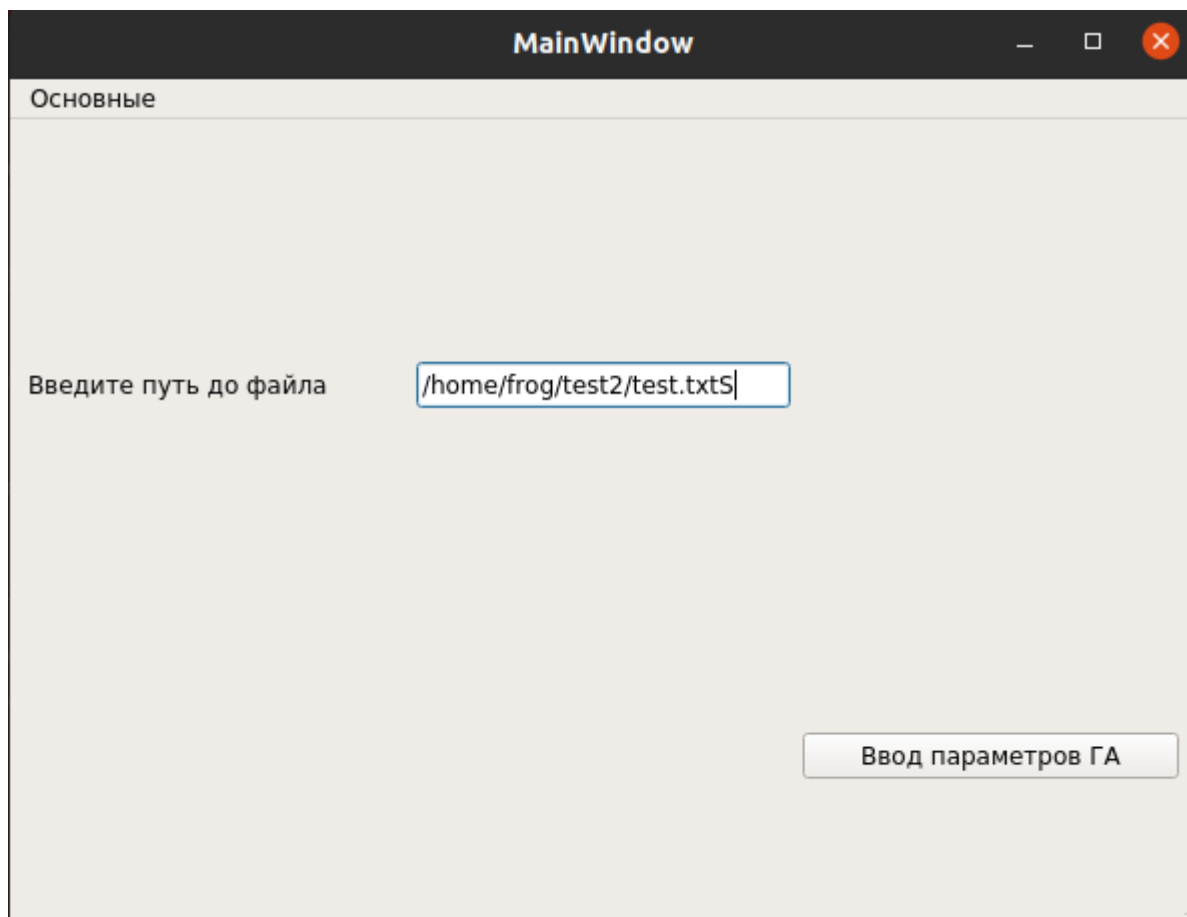


Рисунок 8 — Окно ввода данных с файла

После ввода данных перед пользователем открывается окно ввода параметров ГА, где каждая величина ограничена различными значениями. Например, если пользователь выберет количество особей, участвующих в отборе, большее, чем половина размера популяции, то ввод будет не корректен. Такое условие выбрано для того, чтобы избавиться от преждевременной сходимости ГА.

MainWindow

Основные

Параметры ГА

Seed для генерации популяции ☐ Установить по умолчанию

34

Размер популяции 10

Вероятность мутации 0,70

Количество особей в турнирном отборе 3

Выберите способ окончания алгоритма:

☒ Ограничить число поколений

☐ Поставить число неизменных поколений

10

Переход к решению

Рисунок 9 — Окно ввода параметров ГА

В окне работы алгоритма (рис. 10) пользователю доступны возможности перехода на шаг вперед или назад, а также к конечному решению. Если шаг первый или последний, то пользователю не доступен переход на шаг назад или вперед, соответственно.

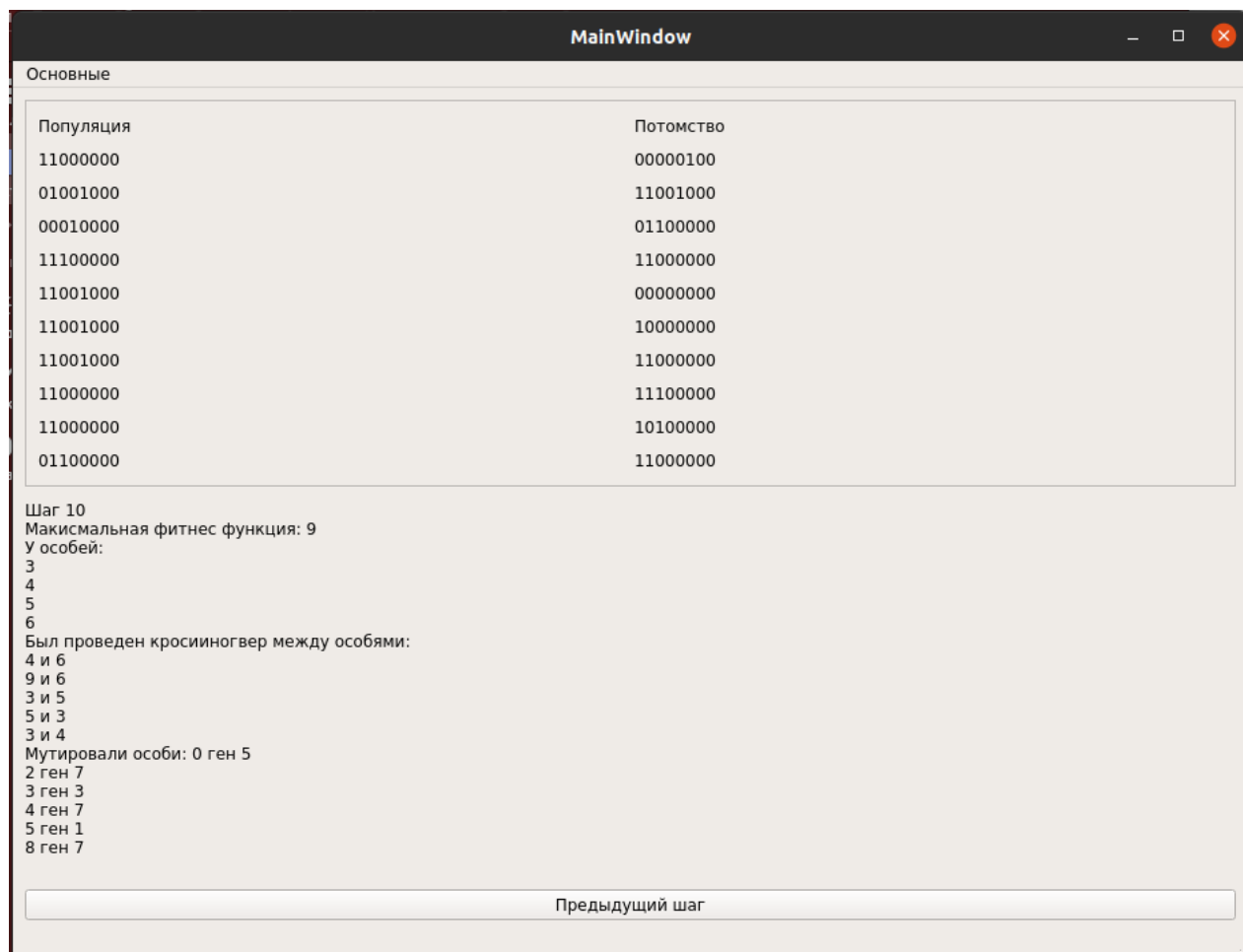


Рисунок 10 — Окно работы ГА

Сходимость алгоритма

Для доказательства сходимости алгоритма рассмотрим пример решения задачи, представленный на рис. 11-14. Проанализировав решение можно сделать вывод, что решение сходится, поскольку было найдено решение, которое лучше первоначального, а также количество особей с таким решений значительно увеличилось.

MainWindow

Основные

Ввод данных о задаче

Вместимость рюкзака

Количество предметов

Введите предметы, где каждый предмет записан в одной строке.
В каждой строке первое число - стоимость, второе - вес.

4	3
5	7
4	2
1	4
6	7
2	2
5	1
3	2

Ввод параметров ГА

Рисунок 11 — Входные данные

MainWindow

Основные

Параметры ГА

Seed для генерации популяции ☐ Установить по умолчанию

34

Размер популяции 10

Вероятность мутации 0,50

Количество особей в турнирном отборе 3

Выберите способ окончания алгоритма:

☒ Ограничить число поколений

☐ Поставить число неизменных поколений

10

Переход к решению

Рисунок 12 — Параметры ГА

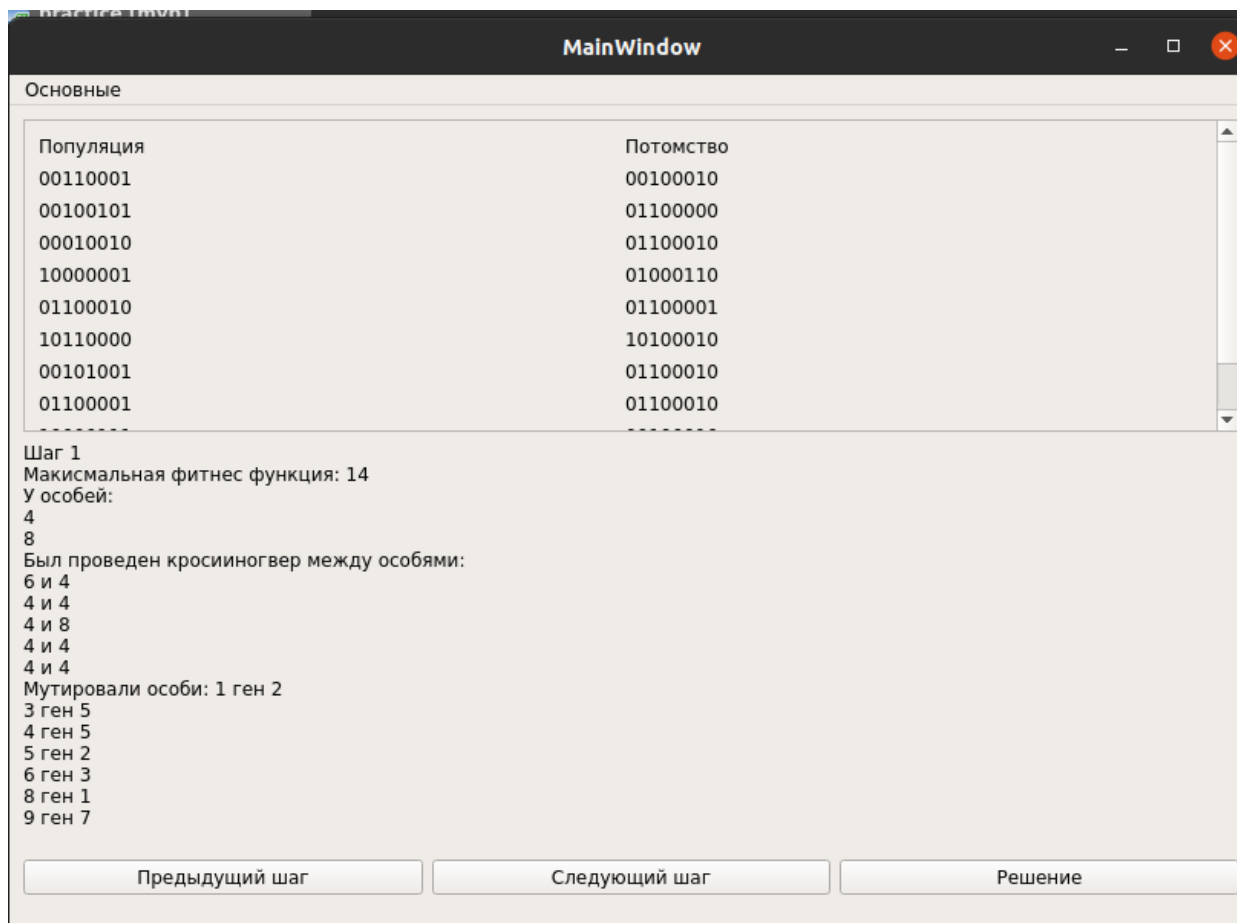


Рисунок 13 — Первый шаг работы ГА

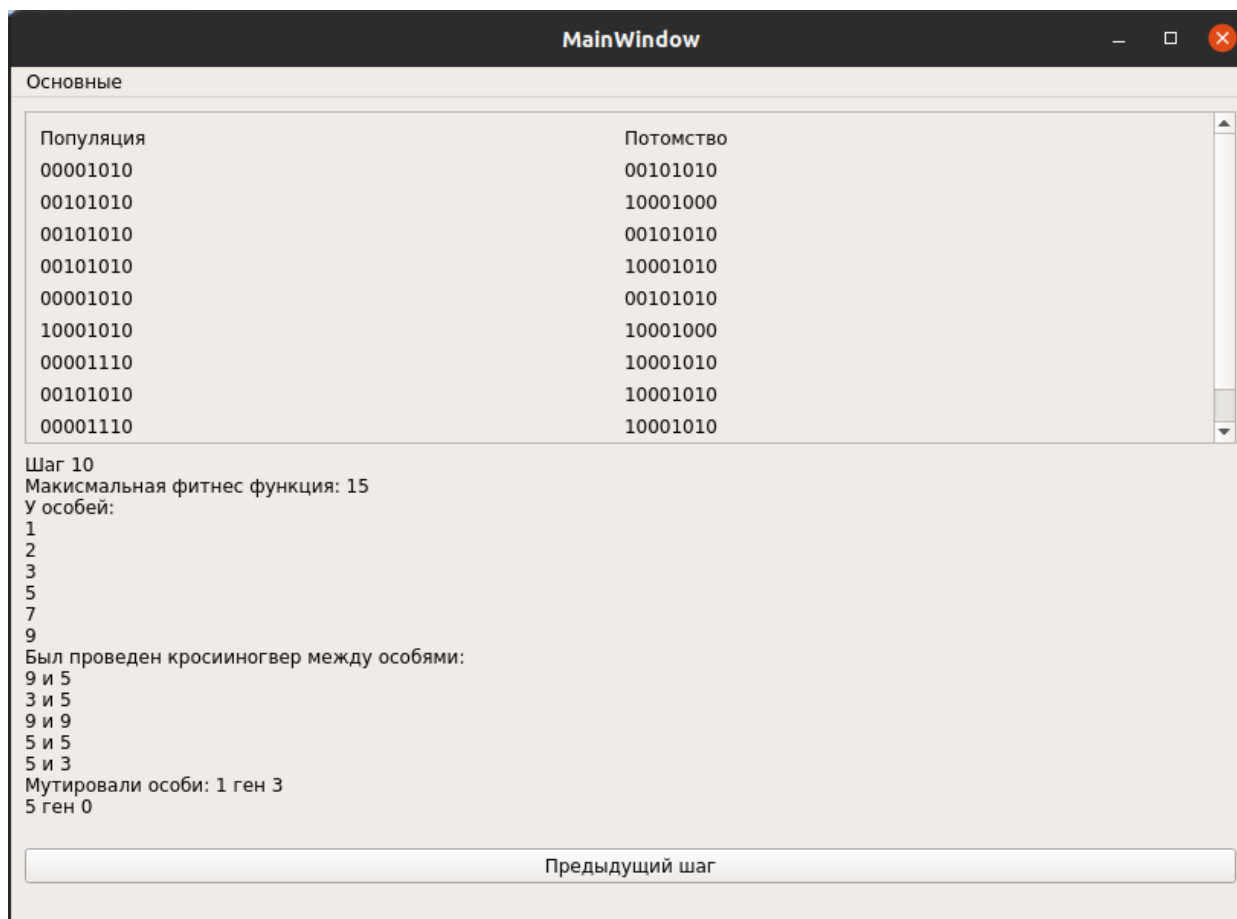


Рисунок 14 — Последний шаг работы ГА

Выводы

В ходе прохождения практики были изучены, освоены и применены основные принципы работы ГА для решения задачи о рюкзаке. Также был изучен фреймворк QT, при помощи которого был реализован GUI.