

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

**ОТЧЁТ**  
**По учебной практике**  
**по направлению “Генетические алгоритмы”**  
**Тема: Задача о рюкзаке**

Преподаватель

Жангиров Т. Р.

Санкт-Петербург

2025

### **Роли в бригаде**

Боривец Савелий - руководитель, разработчик GUI

Канцеров Артемий - программист(реализация основного функционала)

Якушев Пётр - программист(ответственный за мутации и скрещивания)

## Прототип интерфейса

### Окно ввода

Предлагается выбрать несколько способов ввода исходных данных:

- Случайная генерация
- Ввод из txt-файла
- Ввод из GUI

Если будут возникать проблемы, предупреждение снизу поможет разобраться, что не так.

GUI GenAlg

Размер популяции 4

Случайная Генерация Кнопка

Ввод из txt-файла Кнопка

C:\users\data.txt

Ввод из GUI Кнопка

max. размер	40
количество	5
цена	0,4    10    31    5    6,4
вес	3    1    25    6,5    14,1

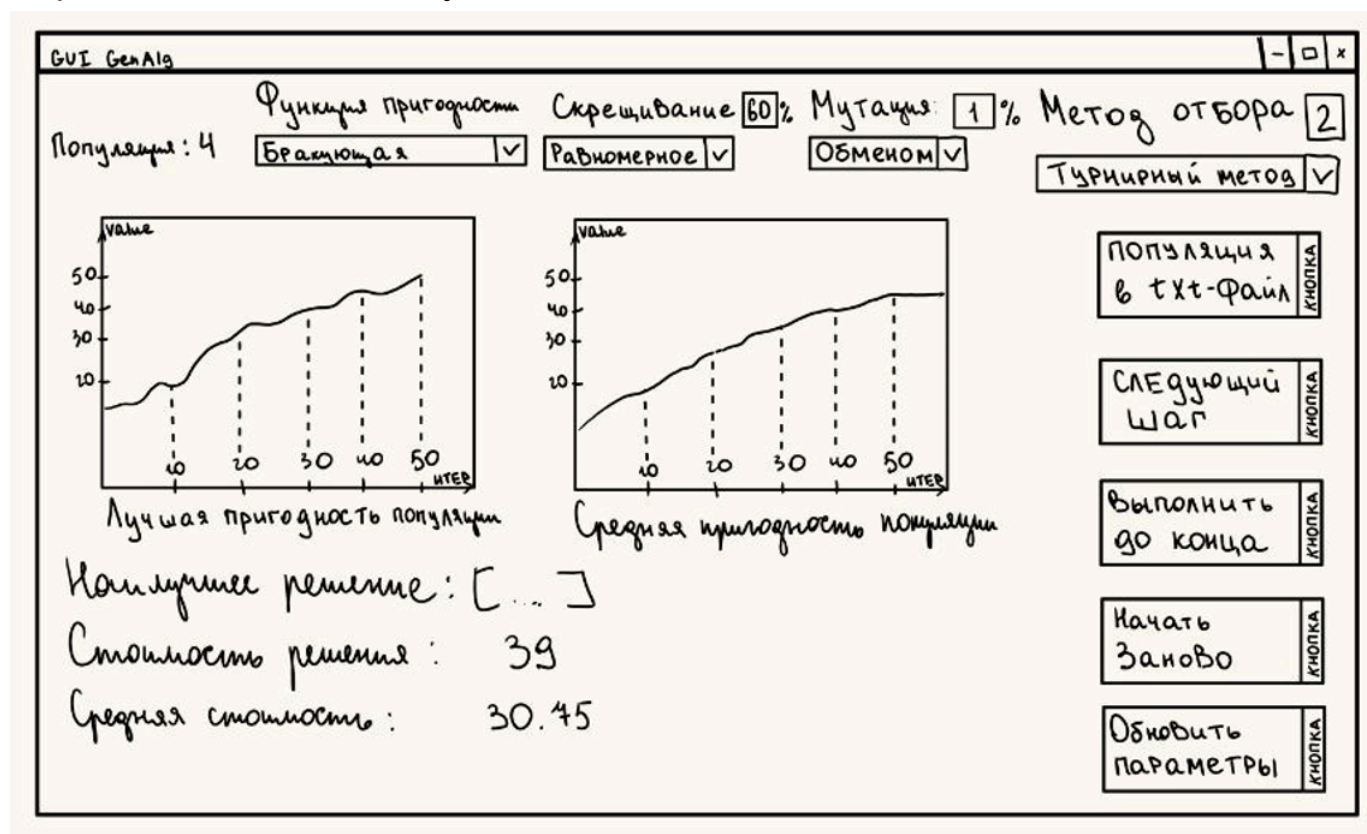
Предупреждение: нет файла по указанному пути / некорректный ввод

## Окно отслеживания работы генетического алгоритма

После того как были введены данные, алгоритм выполняет свою работу.

Можно выбрать различные функции пригодности, скрещивания, мутации и методы отбора, а также выбрать вероятность мутации и скрещивания. Доступно сохранение текущей популяции в txt-файл, выполнение следующего шага, выполнение до конца (до момента критерия окончания процесса), можно начать заново с ранее введенными данными, а также обновить ранее введенные параметры.

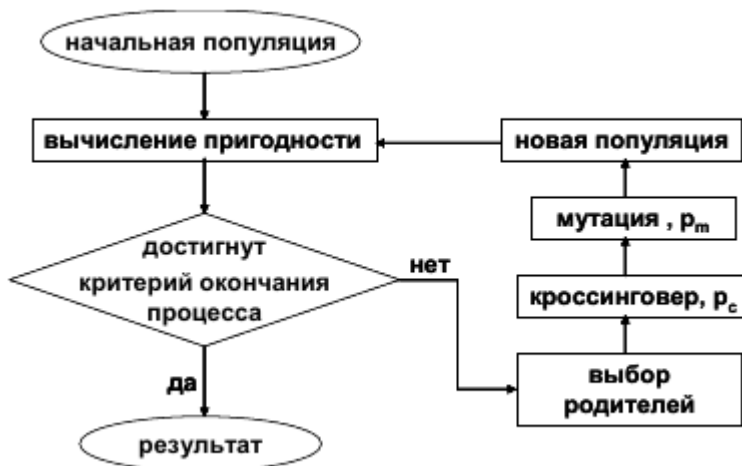
Для наглядности будут реализованы графики лучшей пригодности и средней пригодности популяции, а также данные о наилучшем решении и средней стоимости популяции.



## План решения задачи

Для реализации генетического алгоритмы для решения задачи о рюкзаке необходимо определиться:

- Что будет хромосомой(особью)?
- Как будет работать функция вычисления пригодности?
- Какими методами будет проходить отбор родителей?
- Как будет работать скрещивание?
- Как будет работать мутация?



## Хромосома

Список, представляющий из себя заполненный рюкзак, в который больше не положить любую вещь так, чтобы она не привела к переполнению рюкзака

Геном в хромосоме будет количество  $i$ -ого предмета в упорядоченном по весу списке вещей

## Упорядоченный список вещей

Класс, который будет хранить список упорядоченных по весу предметов, содержать поле максимальной удельной стоимости(для “идеального рюкзака”). Добавляемые предметы будет вставлять на корректные места. Предметы с весом меньшим, либо равным нулю будет исключать

### **Критерий окончания алгоритма**

Критерием окончания алгоритма будет незначительная разница между лучшими стоимостями поколения. Это означает то, что генетический алгоритм пришел к максимуму.

### **Функция вычисления пригодности**

Функция вычисления пригодности(фитнесс-функция)— это функция, которая принимает на входе потенциальное решение проблемы и выдаёт значение, оценивающее его пригодность. В случае задачи о рюкзаке таким значением будет являться общая стоимость вещей в рюкзаке.

Планируется реализовать следующие функции вычисления пригодности:

1. Если общий вес предметов меньше или равен максимальной вместимости рюкзака, то функция вычисления вероятности вернет общую стоимость предметов в рюкзаке. В ином случае она вернет 0, то есть “забракует” решение, из-за чего у него будет гораздо меньше или вообще не будет шансов попасть следующий отбор.
2. К общей стоимости добавляются лишь те предметы, которые при добавлении к текущему общему весу не приведут к превышению максимальной вместимости. Выбор таких предметов будет либо от меньшего к большему, либо случайным образом среди ненулевых элементов.

### **Метод отбора родителей**

В качестве методов отбора планируется создать несколько методов, чтобы посмотреть, какой из них будет наиболее эффективным.

1. Метод отбора по правилу рулетки, или отбор пропорционально приспособленности

В методе рулетки (*roulette-wheel selection*) особи отбираются с помощью  $N$  «запусков» рулетки, где  $N$  — размер популяции. Колесо рулетки содержит по одному сектору для каждого члена популяции. Размер  $i$ -го сектора пропорционален вероятности попадания в новую популяцию  $P(i)$ , вычисляемой по формуле:

$$P(i) = \frac{f(i)}{\sum_{i=1}^N f(i)},$$

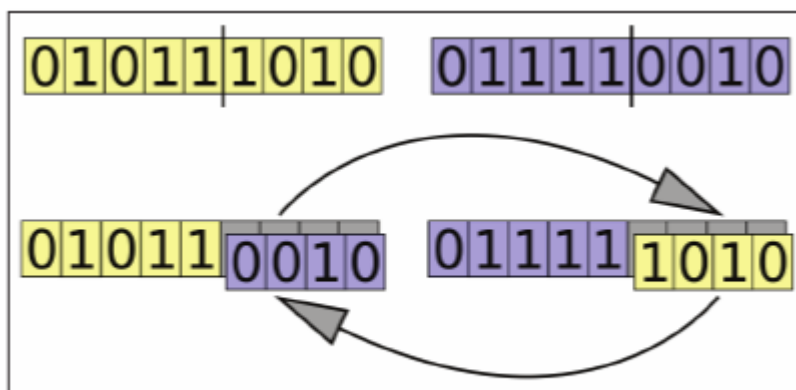
У особей с меньшей приспособленностью будет меньше шансов попасть в следующую популяцию для скрещивания.

2. Турнирный отбор - будет проводиться несколько раундов с фиксированным количеством участников. Среди участников для последующего скрещивания будет взят тот, у которого приспособленность оказалась больше всего. Количество раундов равно размеру популяции. Чем больше участников турнира, тем выше шанс того, что в раунде будут появляться лучшие особи.

### Методы скрещивания

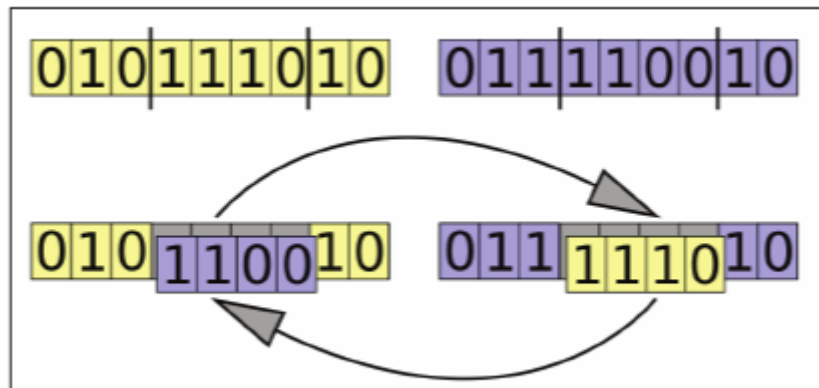
В качестве скрещивания планируется реализовать следующие методы:

1. Одноточечное скрещивание - в хромосоме случайно выбирается позиция - точка скрещивания, затем гены одной хромосомы, расположенные справа от точки скрещивания обмениваются с генами второй хромосомы, расположенные справа от точки скрещивания



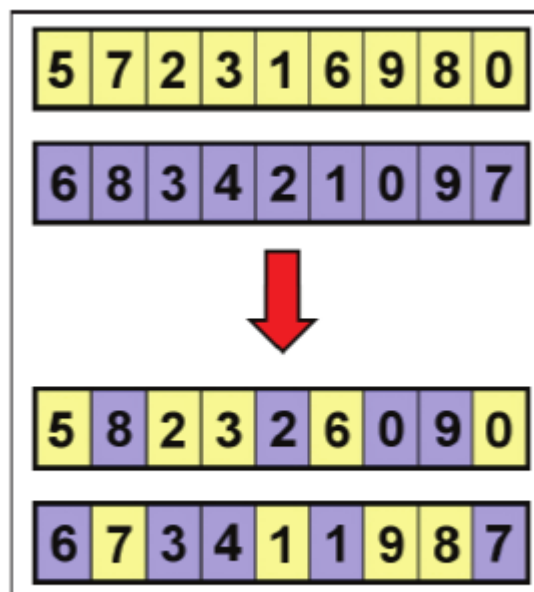
Пример одноточечного скрещивания

2. Двухточечное скрещивание - метод похож на предыдущий, но теперь выбираются две случайные точки скрещивания, и гены хромосом заключенные между этими точками обмениваются



Пример двухточечного скрещивания

3. Равномерное скрещивание - с вероятностью 50% гены обмениваются хромосомами



Пример равномерного скрещивания

### Методы мутации

Мутация – последний генетический оператор, применяемый при создании

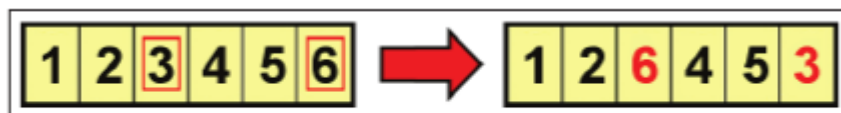
нового поколения.

Мутация будет выполняться с вероятностью 0-2%

Планируется реализовать следующие методы мутации:

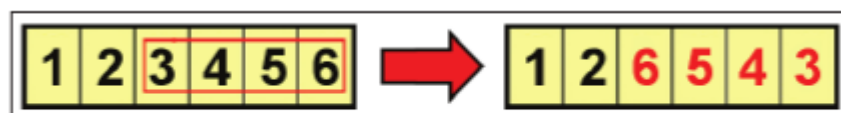


1. Случайное добавление или удаление одной вещи в рюкзак. Если ген будет равен 0(вещи данного типа 0), то вещь будет только добавляться. В ином случае будет выполняться добавление с вероятностью 50%, а в ином случае удаление
2. Мутация обменом - выбирается две позиции, целочисленные значения на данных позициях обмениваются



Пример мутации обменом

3. Мутация обращением - выбирается последовательность в хромосоме, порядок генов в последовательности изменяется на обратный



Пример мутации обращением

### **План реализации алгоритма**

Для начала необходимо реализовать хранение данных, полученных из файла, введенных из GUI или случайной генерации. За это будет отвечать класс DataManager, у которого будут поля-данные, требуемые для решения задачи:

- Количество предметов
- Отсортированный в порядке возрастания веса вектор предметов-структур(информация о предметах будет в виде структур с тремя полями - ценой, весом и удельной стоимостью)
- Максимальная вместимость рюкзака

Данный класс будет иметь следующие методы:

- Метод, добавляющий предмет в список(будет вставлять на нужную позицию)
- Метод, который будет доставать случайный предмет с весом не превышающим значение в аргументах. Он будет нужен для случайного заполнения рюкзака при создании начальной популяции
- Метод, читающий txt-файлы, в которых будет находиться исходные данные задачи
- Сеттеры, которые будут устанавливать полученные данные
- Метод, генерирующий случайные данные задачи

Для хранения данных о хромосоме(особи) будет использоваться класс Backpack с полями:

- Вектор с количеством предметов в рюкзаке(предмет на  $i$ -ой позиции будет иметь цену и вес предмета на  $i$ -ой позиции в списке DataManager)
- Поле, отвечающее за общий вес текущего рюкзака
- Поле, отвечающее за общую цену текущего рюкзака

У него будут методы:

- Добавление  $i$ -ого предмета в рюкзак
- Удаление  $i$ -ого предмета

Функции пригодности будут реализованы в отдельном классе.

Главный метод будет вызывать вспомогательный метод,

возвращающий пригодность решения. Пользователь сможет выбрать функцию пригодности.

Мутации будут реализованы в отдельном классе. Главный метод будет выполнять мутацию с определенной вероятностью, вызывая вспомогательные методы, которые будут принимать объект класса Vaskrack, выполнять операции мутации, а затем возвращать измененный(мутировавший) объект класса Vaskrack. Пользователь сможет выбрать вид мутации, а также ее вероятность.

Будет реализован отдельный класс под скрещивания. Главный метод будет выполнять скрещивание с определенной вероятностью, вызывая вспомогательные методы, которые будут принимать два объекта класса Vaskrack, выполнять операции скрещивания, а затем возвращать два потомка скрещиваемых классов Vaskrack. Пользователь сможет выбрать вид скрещивания, а также вероятность скрещивания.

Работой алгоритма(взаимодействия между классами) будет управлять отдельный класс GenAlg.

Он будет:

1. Генерировать начальную популяцию из  $n$  хромосом.
2. Вычислять для каждой хромосомы ее пригодность.
3. Выбираем пару хромосом-родителей с помощью одного из способов отбора.
4. Проводить скрещивание двух родителей с заданной вероятностью, производя двух потомков.
5. Выполнять мутацию с заданной вероятностью
6. Повторять шаги 3–5, пока не будет сгенерировано новое поколение популяции, содержащее  $n$  хромосом.
7. Повторяем шаги 2–6, пока не будет достигнут критерий окончания процесса.

Для графического интерфейса(GUI) будет использоваться библиотека Qt.