

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

**Факультет информатики, математики и компьютерных наук**

**Программа подготовки бакалавров по направлению  
01.03.02 Прикладная математика и информатика**

*Пекерская Даниэла Михайловна*

**КУРСОВАЯ РАБОТА**

Генетический алгоритм для решения задачи о вершинном покрытии  
графа

Научный руководитель:

Старший преподаватель

---

Т.С. Бабкина

Нижний Новгород, 2024

# Содержание

Введение	3
1 Постановка задачи	4
1.1 Математическая модель . . . . .	4
1.2 Применимые подходы . . . . .	4
1.3 Обзор литературы . . . . .	5
2 Решение задачи . . . . .	9
2.1 Описание генетического алгоритма как общего подхода к решению задачи оптимизации	9
2.2 Применение генетического алгоритма к решению задачи о вершинном покрытии графа	12
3 Результаты экспериментов . . . . .	15
Заключение	16
Список литературы	17

## Введение

Задача о вершинном покрытии графа - одна из фундаментальных задач теории графов. Ее используют в множестве различных областей - задача находит своё применение и в вопросах транспорта (при оптимальном планировании размещения остановок), и в таких развивающихся науках как биоинформатика, при планировании распределения точек доступа Wi-Fi задача может помочь определить самые эффективные позиции для максимального покрытия сигналом, и в многих других. Эффективное решение данной задачи позволяет оптимизировать распределение ресурсов и повысить эффективность разнообразных процессов.

Традиционные методы поиска решений для задачи о вершинном покрытии графа, основанные на полном переборе вариантов, требуют значительных временных затрат, особенно для больших графов. Альтернативным подходом является использование генетических алгоритмов (ГА) - эвристических методов, вдохновленных процессами эволюции в природе.

Основное преимущество применения ГА для решения задачи о вершинном покрытии заключается в их способности находить "вполне" удовлетворительные решения за меньшее время по сравнению с другими алгоритмами поисковой оптимизации. Хотя ГА не гарантируют обнаружения глобального оптимума за полиномиальное время, они эффективно выявляют решения, близкие к оптимальным, что делает их ценным инструментом для практических приложений.

Цель данной работы: исследование возможностей ГА для решения задачи о вершинном покрытии графа.

Чтобы достичь этой цели необходимо решить следующие задачи:

- 1) изучить применения генетических алгоритмов к решению задач дискретной оптимизации;
- 2) применить генетический алгоритм к задаче о вершинном покрытии графа;

- 3) определить, как качество работы алгоритма зависит от характеристик графа и параметров самого алгоритма.

## 1. Постановка задачи

### 1.1 Математическая модель

Рассматривается неориентированный граф  $G=(V,E)$ , где  $V$ -множество вершин,  $E$  - множество ребер. Число вершин равно  $n$ ; число рёбер равно  $m$ .

Требуется найти минимальное покрытие данного графа: подмножество вершин  $P \subset V$ , такое, что любое ребро  $e \in E$  инцидентно какой-либо вершине  $v \in P$ , и  $|P|$  минимально.  $|P|$ -число элементов в данном множестве.

Математическая модель включает:

- Переменные  $x_i = \begin{cases} 1 & \text{если вершина } i \text{ входит в покрытие,} \\ 0 & \text{в противном случае,} \end{cases}$

где  $i \in \{1, \dots, n\}$

- Целевую функцию, подлежащую минимизации

$$f(x) = \sum_{i=1}^n x_i \quad (1.1)$$

- Ограничения:

Каждое ребро должно быть покрыто:

$$\forall (u, v) \in E: x_u + x_v \geq 1 \quad (1.2)$$

$$x_i \in \{0,1\} \forall i \in V \quad (1.3)$$

### 1.2. Применимые подходы

К данной задаче можно применить такие подходы, как:

1. Полный перебор;
2. Алгоритм жадной аппроксимации;
3. Генетические алгоритмы;
4. Метод пересечения ребер;
5. Решение с помощью систем квадратичных уравнений.

При решении задачи о вершинном покрытии графа полным перебором рассматриваются все возможные комбинации вершин графа, затем каждая комбинация проверяется на соответствие условию вершинного покрытия, после чего находится минимальный размер вершинного покрытия.

Решение с помощью алгоритма жадной аппроксимации: поочерёдно рассматриваются вершины графа, на каждой итерации выбирается вершина, покрывающую наибольшее количество ещё не покрытых рёбер. Это алгоритм находит простое и быстрое, гарантирующее что найденное вершинное покрытие будет не более чем в два раза больше оптимального вершинного покрытия. Однако, минусом такого подхода является то, что алгоритм не всегда даёт оптимальное решение.

Решение с помощью генетических алгоритмов: имитируется эволюционный процесс для поиска вершинного покрытия с небольшим количеством вершин. Этот метод особенно может использоваться для поиска хороших решений в больших и сложных графах.

Решение с помощью метода пересечения ребер: происходит поиск вершин, которые пересекаются с наибольшим количеством еще не покрытых ребер, и они добавляются в вершинное покрытие. Этот подход считается простым и эффективным, он часто используется для поиска приближений к оптимальному вершинному покрытию. Однако этот метод не гарантирует оптимальное решение.

Решение с помощью систем квадратичных уравнений: в этом методе задача о вершинном покрытии формулируется как система квадратичных уравнений и решается с помощью методов оптимизации.

Этот подход может быть использован для поиска оптимального решения в небольших графах, однако он является вычислительно затратным для больших графов.

### **1.3 Обзор литературы**

В книге [1] рассматриваются математические основы генетических алгоритмов, а также: компьютерная реализация генетического алгоритма, некоторые приложения генетических алгоритмов. Кроме того, в книге автор также приводит пример реализации простого генетического алгоритма (SGA) на языке программирования Pascal.

В статье [2] рассматривается решение задачи о вершинном покрытии графа на малых графах и ее особенности.

В статье [3] представлен подход к решению проблемы минимального вершинного покрытия в 3-регулярных графах. Автор статьи рассматривает специально построенные планарные 3-регулярные графы и демонстрирует, как эффективно определить их минимальные наборы покрытий и минимальное количество вершин в них. Однако, его методы актуальны только для графов, описанных выше.

В работах [4] и [8] рассматривается проблема “Вечного покрытия вершины”- вариант проблемы покрытия вершины графа, который можно использовать как динамическую игру между двумя игроками (Атакующий и Защитник) с бесконечным количеством оборотов. В этой игре защитник размещает охранников на графе, чтобы защитить его от атак, а атакующий пытается найти уязвимости. Цель — найти минимальное число охранников для успешной защиты графа. Минимальное количество охранников, с которыми защитник имеет выигрышную стратегию называется *Enteral Vertex Cover number* для графа  $G$ .

В статье [5] авторами также рассматривается проблема “Вечного покрытия вершины”. Однако, в отличии от работ [4] и [8] в статье рассматриваются только двудольные и полные двудольные графы.

В работе [6] производится анализ задачи вершинного покрытия в планарных двухчастных графах с 3 ребрами.

В статье [7] рассматривается алгоритм для решения задачи покрытия вершин с минимальным весом в больших графах.

В книге [9] излагаются основы информационной технологии, позволяющей сводить классические задачи дискретной оптимизации (комбинаторные задачи о ранце, коммивояжере, покрытии и разбиении) к задаче поиска на дискретном множестве кодировок. Рассматриваются основные принципы, типовые структуры и механизмы предлагаемого популяционно-генетического подхода к решению задач поиска с помощью генетических

методов. В книге описаны основы генетического поиска и приведены конкретные примеры, в которых большое внимание уделяется вычислительной реализации генетических методов.

В работе [10] рассмотрены основные стратегии, принципы и концепции направления «Генетические алгоритмы». Описаны фундаментальные основы генетических алгоритмов и эволюционного моделирования. Проанализированы архитектуры генетического поиска и модели генетических операторов. Приведены конкретные примеры решения основных задач оптимизации на основе генетических алгоритмов и дано большое число контрольных вопросов и упражнений.

В работе [11] рассматривается применение методологии многоуровневого программирования к задачам стандартизации и размещения, исследование свойств двухуровневых задач стандартизации, выделение классов задач, допускающих эффективное решение, разработка подходов к построению нижних оценок оптимальных значений целевых функций рассматриваемых задач. В работе описываются методы математического программирования, применяемые к решению этих задач, в частности, методы динамического, линейного, дискретного программирования.

В статье [12] рассматривается метод, использующий для решения задачи системы квадратичных уравнений. Этот способ решения позволяет достигать высокой степени распараллеливания операций. Проводится прогнозируемая выборка пар вершин с наиболее оптимальной общей степенью, на основании этой выборки составляется система квадратичных уравнений, при решении которой возвращается результат, являющийся наименьшим покрытием заданного графа.

В книге [13] рассмотрены генетические алгоритмы, широко применяемые в настоящее время для решения задач оптимизации. Кроме того, в книге описываются стандартные функции универсального пакета MATLAB 7.0.1, предназначенные для решения задач оптимизации с помощью генетических алгоритмов.



В статье [14] рассматривается задача о вершинном покрытии на гиперграфах. Гиперграфом называется обобщение графа, в котором каждым ребром могут соединяться не только две вершины, но и любые подмножества множества вершин. Результаты статьи: чем больше количество вершин в графе, тем жадный алгоритм находит более оптимальные решения в отличие от ленивого алгоритма.

## 2. Решение задачи

### 2.1 Описание генетического алгоритма как общего подхода к решению задачи оптимизации

Генетические алгоритмы представляют собой метод оптимизации, моделирующий принципы естественного отбора и генетики для поиска решений в пространстве задач. Алгоритм оперирует популяцией решений, представленных как "особи", и использует операторы скрещивания, мутации и селекции для поиска оптимального решения.

Особи, представленные как хромосомы, состоят из отдельных элементов - генов, которые являются двоичными числами. Каждая особь соответствует какой-то допустимой точке рассматриваемой задачи.

Введем обозначения:

$G(0)$  - начальная популяция, сгенерированная случайно;

$P(0)$  - ее потомки;

$G(1)$  - новая популяция, отобранная из  $G(0) \cup P(0)$ ;

$G(i)$  популяция,  $P(i)$  - ее потомки; из  $G(i) \cup P(i)$  производится отбор в популяцию  $G(i+1)$ .

Основные этапы работы генетического алгоритма:

#### 1. Инициализация популяции:

Начальная популяция особей генерируется случайным образом

#### 2. Оценка приспособленности и допустимости:

Для каждой особи вычисляется ее приспособленность на основе целевой функции. Если значение целевой функции стремится к минимуму, то значение функции приспособленности этой особи устремляется к максимуму и наоборот соответственно. Контролируется соответствие полученной особи допустимой точке. (Особь считается допустимой, если все ребра графа покрыты). Если точка недопустима, то равновероятен выбор из 2х вариантов: значению функции приспособленности присваивается значение 0.5 или особь корректируется до того момента, пока не начнет соответствовать допустимой точке.

### 3. Выбор родительской пары:

Процедура выбора особей для скрещивания может быть осуществлена одним из следующих методов:

- 1) Панмиксия - каждой особи из популяции поставить в соответствие случайно выбранную особь;
  - 2) Инбридинг - каждой особи из популяции поставить в соответствие наиболее близкую к ней особь;
  - 3) Аутбридинг - каждой особи из популяции поставить в соответствие наиболее далекую от нее особь;
  - 4) Селекция - выбрать  $N$  особей чья приспособленность не ниже некоторого порогового значения;
  - 5) Турнирный отбор - выбрать  $N$  особей победителей в турнирах;
  - 6) Пропорциональный отбор - выбрать  $N$  особей с помощью рулетки.
- ### 4. Скрещивание (кроссинговер):

Стандартный подход к скрещиванию заключается в обмене частями хромосом между родителями для производства потомства с комбинированными характеристиками.

Выбирается случайная позиция  $k$ ,  $0 \leq k \leq n-1$  ( $n$  – число вершин в графе), которая делит хромосому каждого родителя на 2 части. Затем эти 2 части каждого родителя меняются местами.

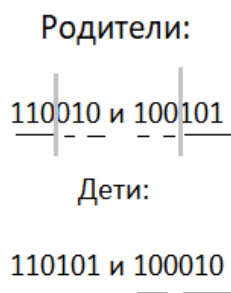


Рис. 1. Скрещивание.

### 5. Мутация:

Введение случайных изменений в хромосомы особей для исследования новых решений в пространстве поиска.

Выбирается случайная позиция  $k$ , затем элемент хромосомы с индексом  $k$  заменяется на обратный ему элемент.

Особь: 111110

Случайный индекс для изменения:  $k=3$

элемент с индексом  $k=3$  равен 1 =>  
после мутации элемент изменяет свое  
значение на 0

Полученная особь: 111010

Рис. 2. Пример мутации.

#### 6. Сохранение чемпиона:

Сохранение лучших особей в следующее поколение для сохранения хороших характеристик из поколения в поколение.

#### 7. Отбор

Отбор в новое поколение на основании характеристик особей.

- Пропорциональный отбор.  $N$  запусков рулетки;
- Канонический ГА - потомки заменяют родителей;
- Отбор по возрасту - удаление наиболее ранних особей;
- Турнирный отбор - остаются лишь  $N$  лучших особей;
- Сочетание разных способов.

#### 8. Повтор процесса эволюции (отбора, скрещивания, мутации) в несколько итераций до достижения критерия остановки.

### 2.2 Применение ГА к решению задачи о вершинном покрытии графа

При решении задачи о вершинном покрытии графа каждая хромосома  $X = (X_1, X_2, \dots, X_n)$  соответствует некоторому покрытию  $x = (x_1, x_2, \dots, x_n)$ .

Каждый ген  $X_i$  принимает значение 0 или 1:

$$X_i = x_i = \begin{cases} 1, & \text{если вершина входит в покрытие,} \\ 0, & \text{если вершина не входит в покрытие,} \end{cases} \quad \text{где } n - \text{число вершин в графе.}$$

Таким образом, хромосома кодирует набор переменных, удовлетворяющих ограничению (1.3).

Хромосома (особь) является допустимой, если она соответствует допустимому покрытию, т.е. удовлетворяет ограничению (1.2). В противном случае она является недопустимой. Целевая функция для каждой точки задачи представляет собой число вершин, входящих в покрытие. Тогда если  $X$  - хромосома, соответствующая покрытию  $x$ , то целевая функция  $f(x)=x_1+x_2+\dots+x_n$ .

В задаче целевая функция подлежит минимизации. Функция приспособленности особи в ГА, в свою очередь, определяется следующим образом:

$$\text{Fitness\_Function}(X) = \frac{10n}{1 + f(x)}$$

и подлежит максимизации.

### Пример.

Рассмотрим граф с 6 вершинами и ребрами  $\{(0,1), (0,2), (0,3), (0,4), (1,2), (2,3), (2,5), (4,5)\}$

а) Рассмотрим особь с генотипом (101001). Она соответствует покрытию, в которое входят вершины 0, 2, 5. Покрытие является допустимым, так как все ребра покрыты. значение целевой функции  $F(x)=3$ , функция приспособленности

$$\text{Fitness\_Function} = \frac{10 \cdot 6}{1 + 3} = 15$$

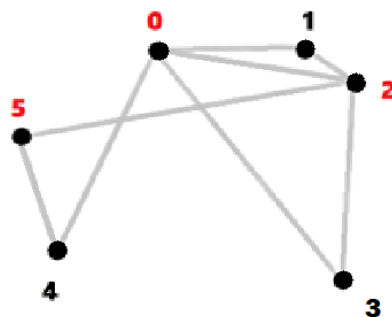


Рис. 3. Допустимое покрытие (красным цветом обозначены вершины, входящие в покрытие)

б) Рассмотрим особь с генотипом (110011). Она соответствует покрытию, в которое входят вершины 0, 1, 4, 5. Покрытие не является допустимым так как не покрыто ребро (2, 3).

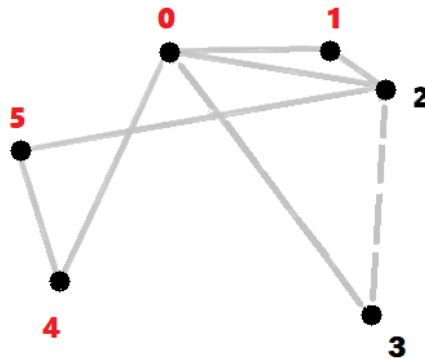


Рис. 4. Недопустимое покрытие (красным цветом обозначены вершины, входящие в покрытие, черным – не входящие в покрытие; пунктиром обозначены непокрытые ребра)

Если особь оказывается недопустимой, то с вероятностью  $\frac{1}{2}$  либо функции приспособленности этой особи присваивается значение 0.5 либо особь изменяется до тех пор, пока не станет допустимой. При этом в хромосому случайным образом добавляются единицы

В предыдущем примере возможны два варианта действий:

- Присвоить этой особи приспособленность

$$Fitness\_Function(X) = 0,5 ;$$

- Скорректировать особь, добавив в покрытие вершину 0, и получить особь (111010) с целевой функцией  $f(x)=4$  и приспособленностью  $Fitness\_Function(X) = 12$ .

Для решения этой задачи используются следующие генетические операторы:

- 1) Метод стандартного кроссинговера;
- 2) Стандартная мутация;
- 3) Метод панмиксии для выбора родительской пары;

4) Пропорциональный отбор для формирования следующего поколения.

#### 4. Результаты экспериментов

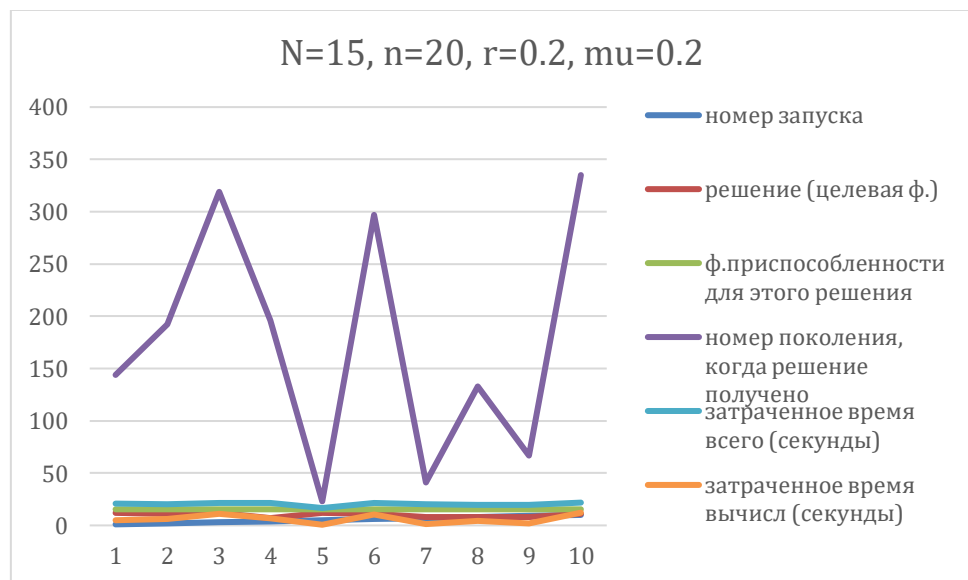
Алгоритм был реализован на языке Python 3.12.1. Расчеты проводились на компьютере Honor Magic Book 14.

Работа генетического алгоритма рассматривалась на графах с различным количеством вершин и ребер. Были созданы графы с количеством вершин  $n = 20, 25, 30, 50$ . В качестве параметров ГА в работе рассматриваются:

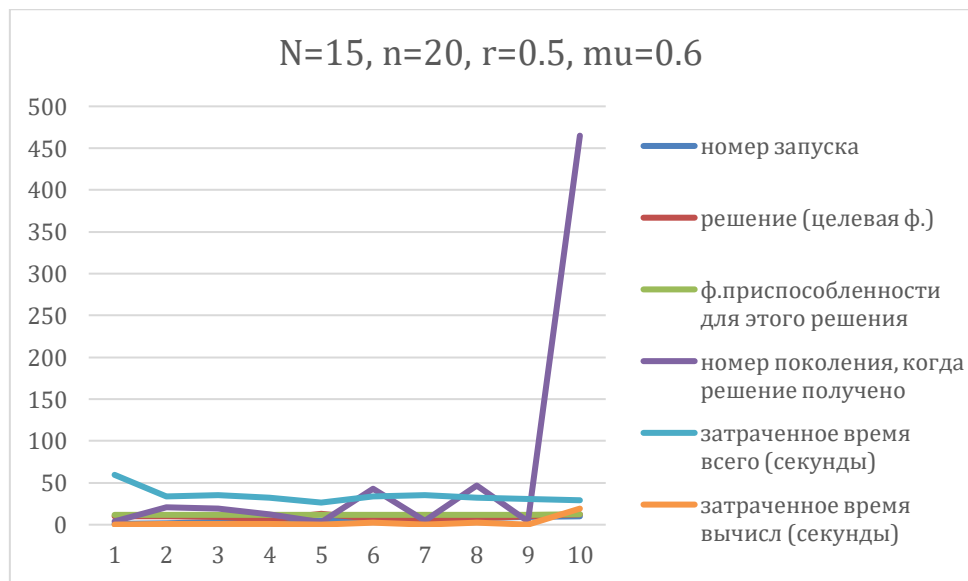
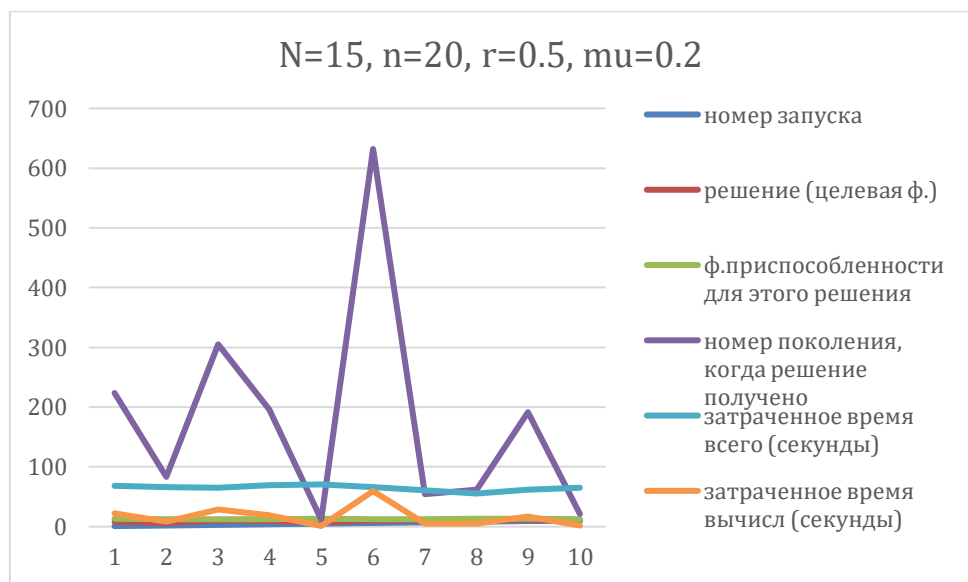
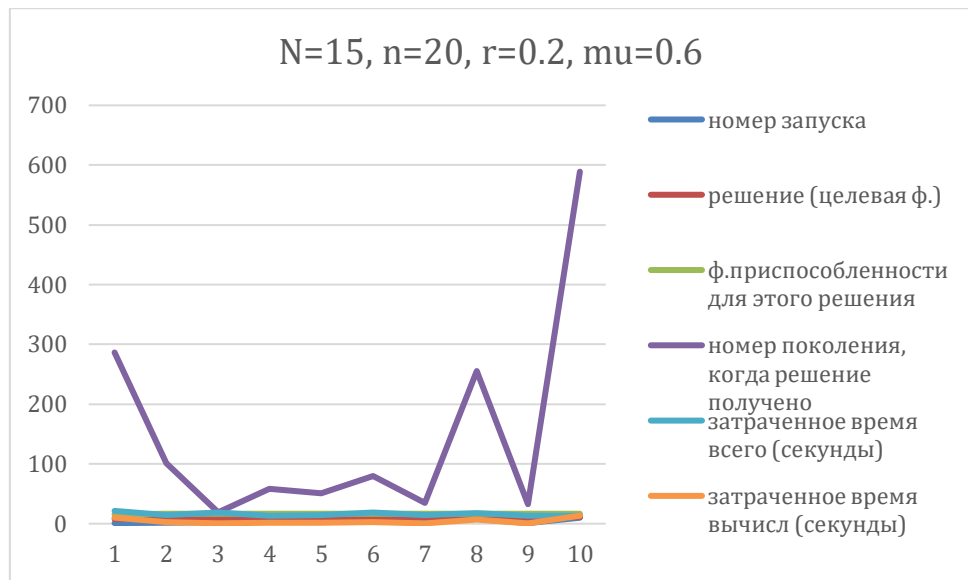
- вероятность мутации;
- размер популяции;
- число поколений.

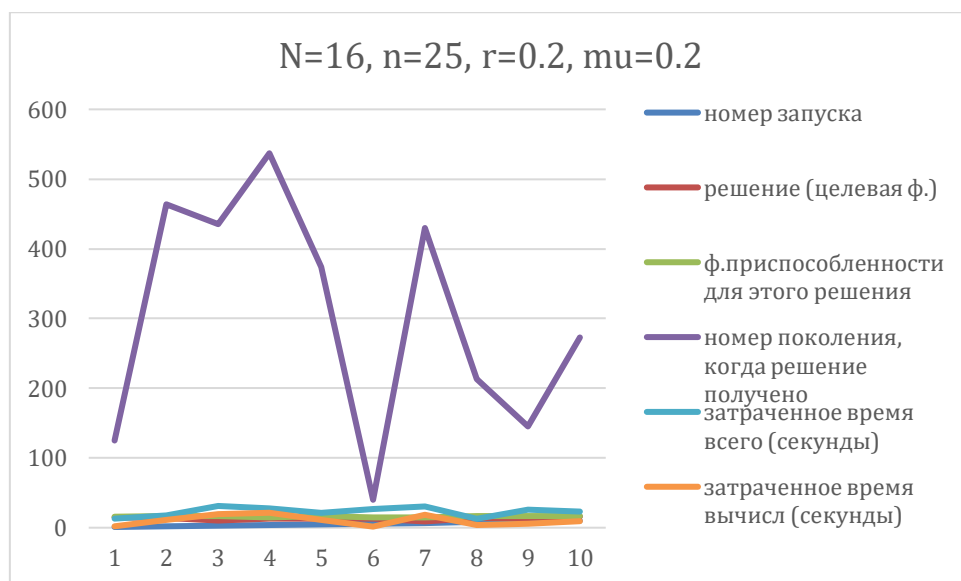
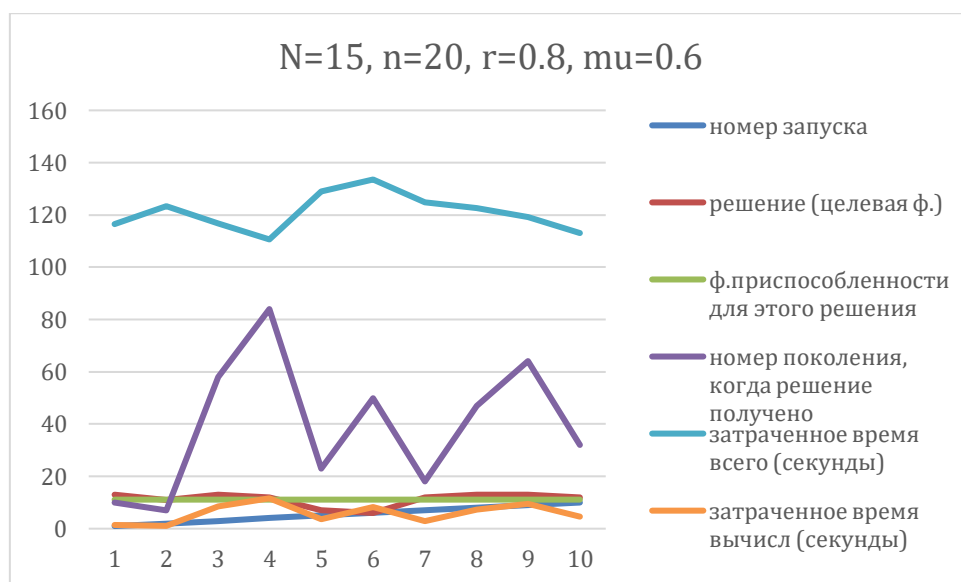
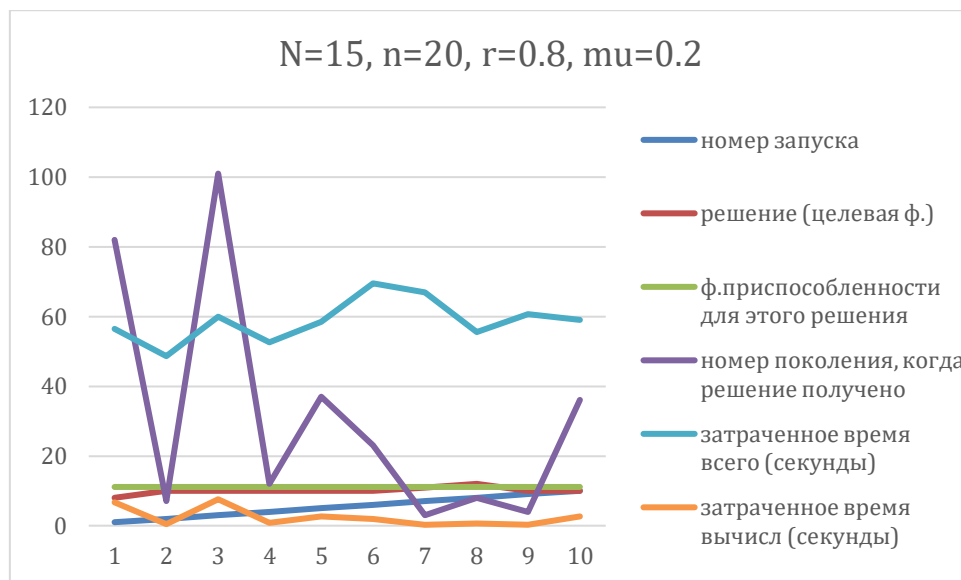
На каждом графе алгоритм испытывался десять раз. Результаты работы алгоритма сравнивались между собой.

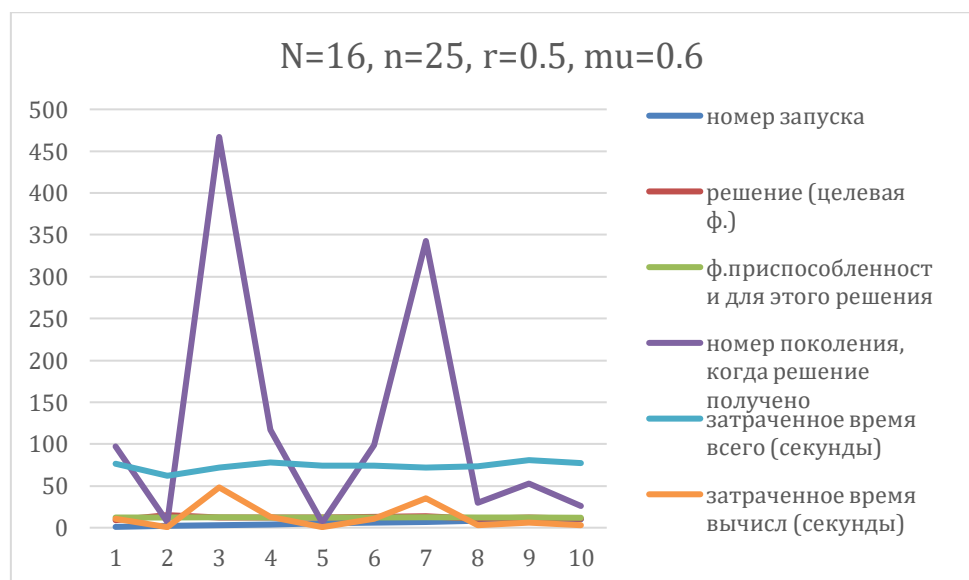
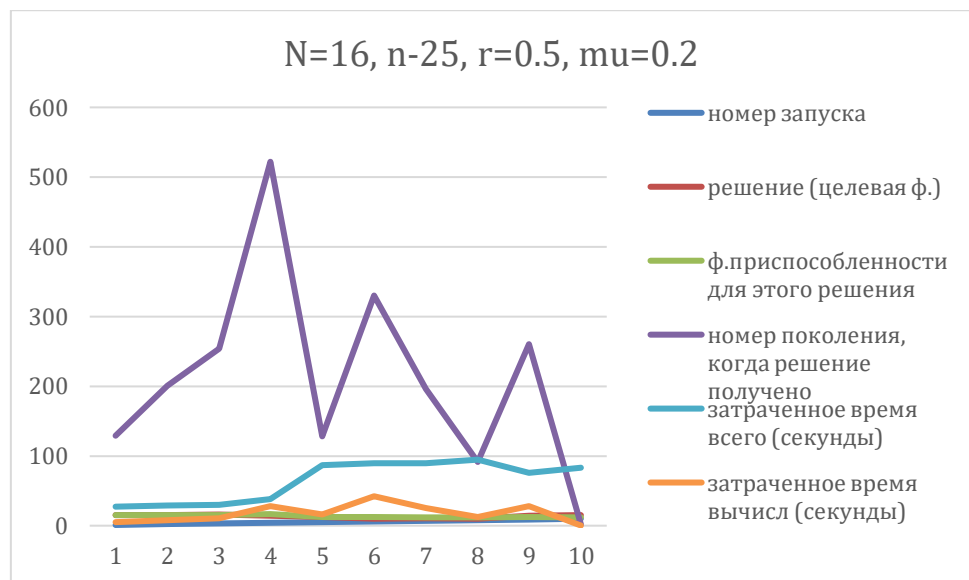
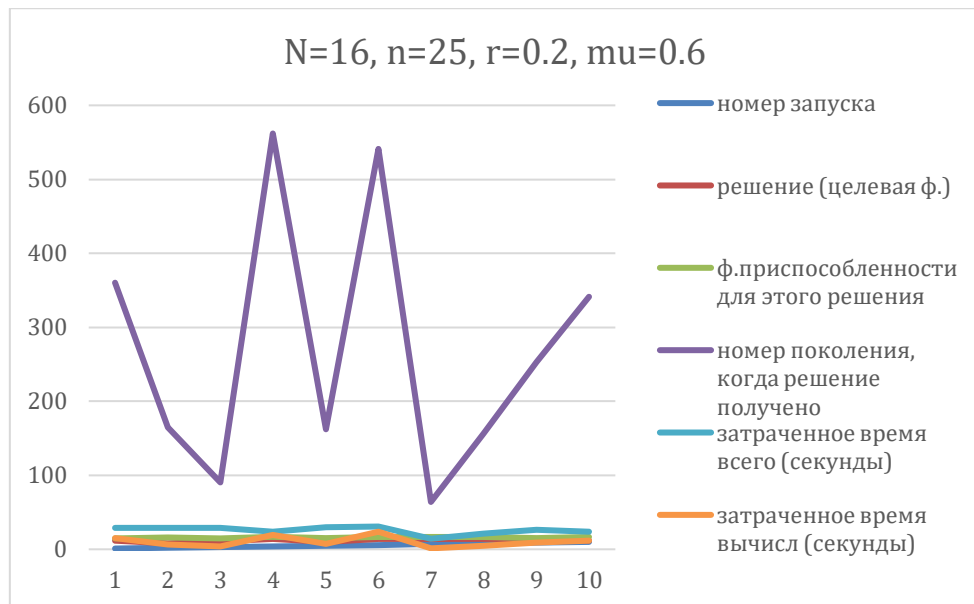
Ниже представлены результаты экспериментов.

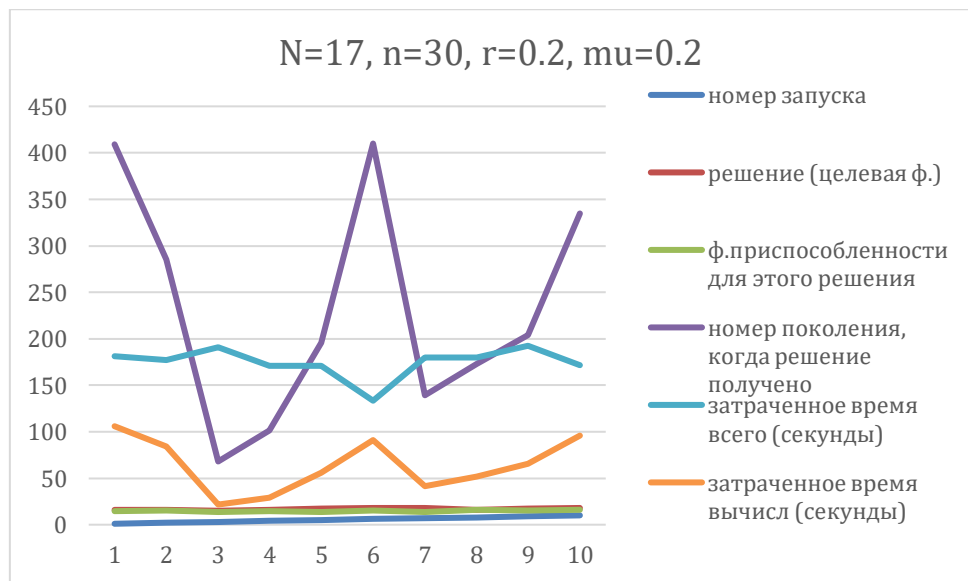
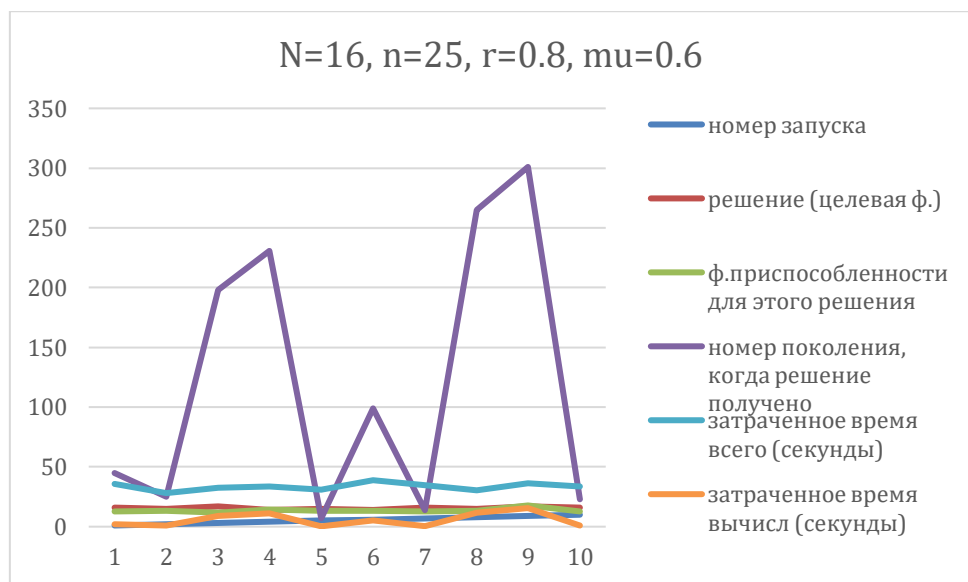
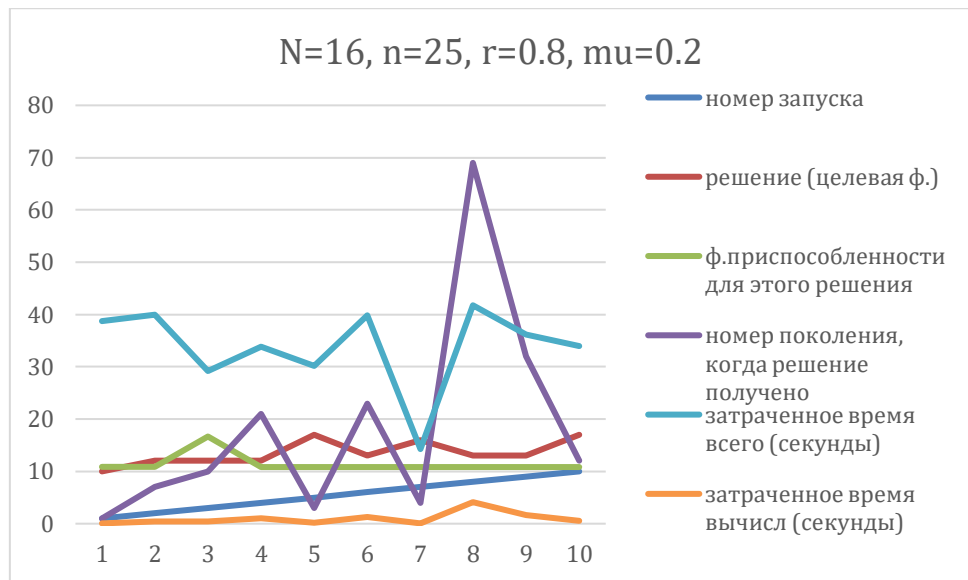


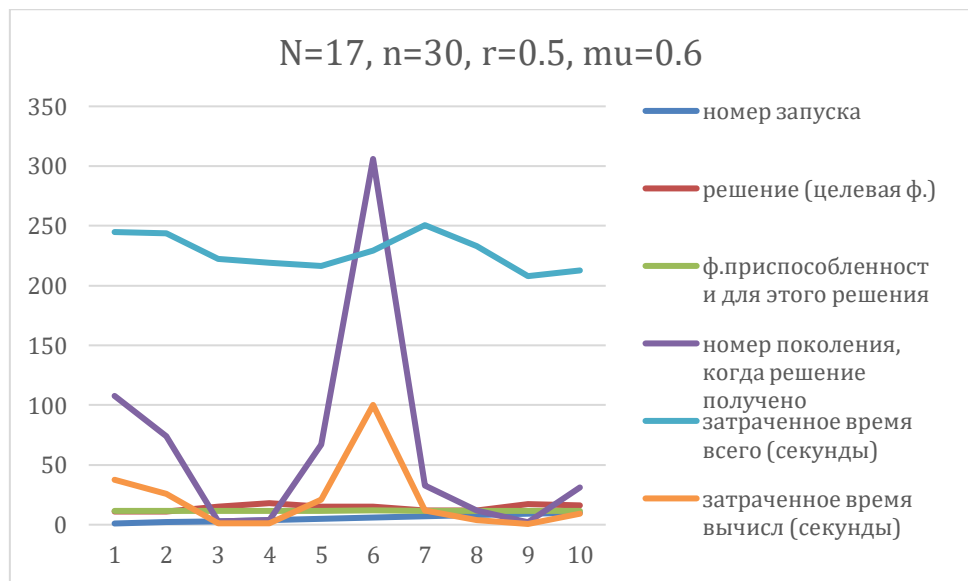
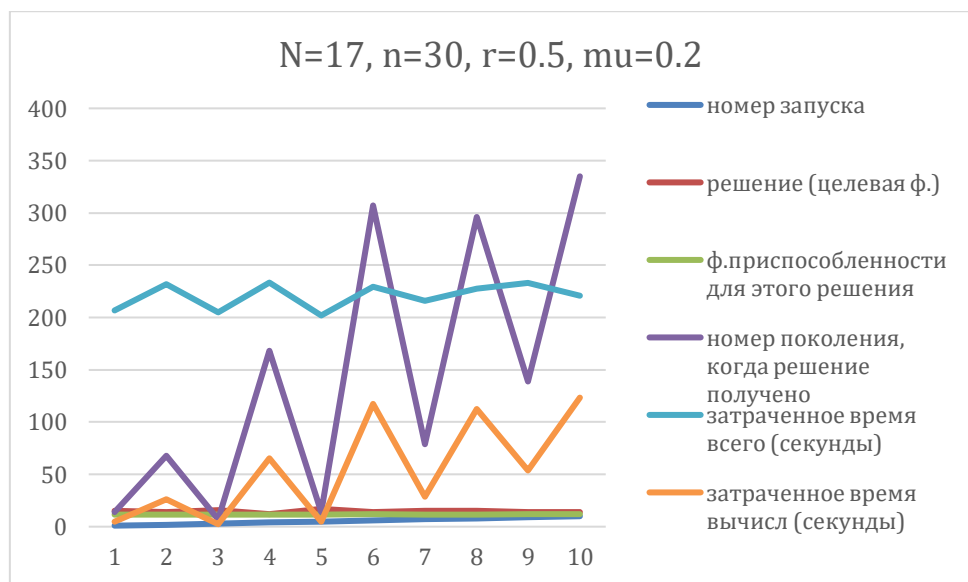
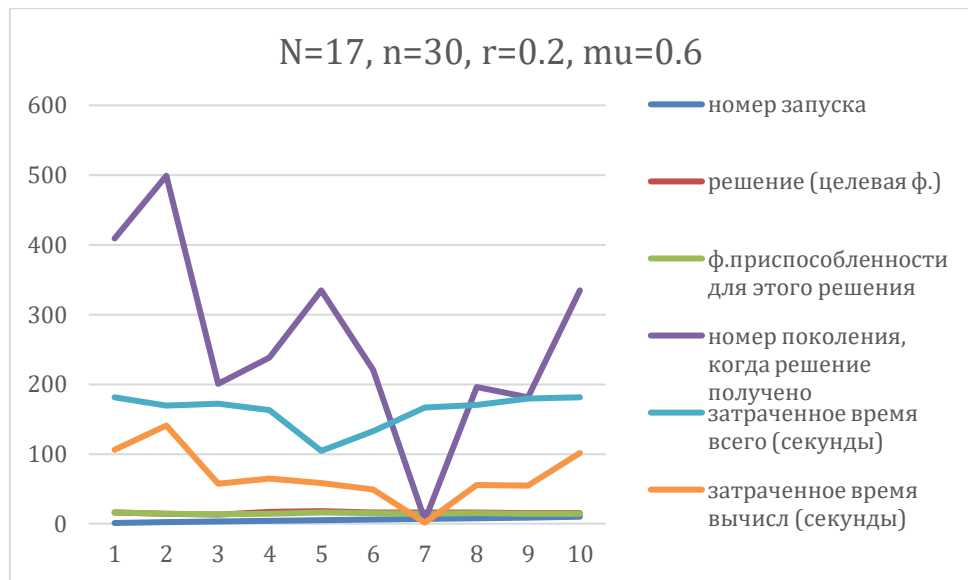


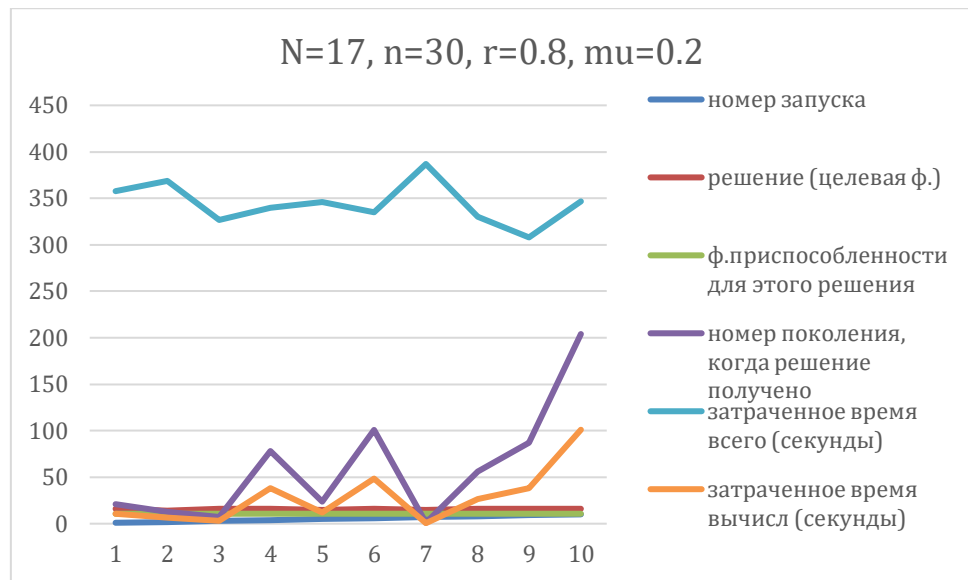












Группа графиков 1. Результаты экспериментов.

### Итоговая таблица

n	n=20					
r, $\mu$	r=0.2, $\mu=0.2$	r=0.2, $\mu=0.6$	r=0.5, $\mu=0.2$	r=0.5, $\mu=0.6$	r=0.8, $\mu=0.2$	r=0.8, $\mu=0.6$
номер запуска	4	8	2	7	1	6
решение (целевая ф.)	7	11	6	6	8	6
ф.приспособленности для этого решения	15,38461538	16,66666667	12,5	11,76470588	11,11111111	11,11111111
номер поколения, когда решение получено	196	255	83	5	82	50
затраченное время (секунды)	6,93	7,40	7,88	0,25	6,62	8,35
n	n=25					
r, $\mu$	r=0.2, $\mu=0.2$	r=0.2, $\mu=0.6$	r=0.5, $\mu=0.2$	r=0.5, $\mu=0.6$	r=0.8, $\mu=0.2$	r=0.8, $\mu=0.6$
номер запуска	1	6	2	9	1	4
решение (целевая ф.)	15	15	11	12	10	14
ф.приспособленности для этого решения	15,625	15,625	12,5	11,764	10,87	14,286
номер поколения, когда решение получено	129	225	137	203	1	231
затраченное время (секунды)	5,03	7,09	47,25	42,14	0,38	11,08
n	n=30					
r, $\mu$	r=0.2, $\mu=0.2$	r=0.2, $\mu=0.6$	r=0.5, $\mu=0.2$	r=0.5, $\mu=0.6$	r=0.8, $\mu=0.2$	r=0.8, $\mu=0.6$
номер запуска	3	3	4	2	2	8
решение (целевая ф.)	15	13	12	11	14	15
ф.приспособленности для этого решения	13,63636364	13,63636364	11,53846154	11,53846154	10,71428571	13,33
номер поколения, когда решение получено	68	201	168	74	13	265
затраченное время (секунды)	66,30	57,59	63,35	25,76	6,85	11,46

Табл.1. Результаты экспериментов.

Основные выводы работы заключаются в следующем:

- Генетический алгоритм способен эффективно находить приближение наименьшего вершинного покрытия в графах с разными характеристиками.
- Увеличение вероятности существования ребер приводит к снижению эффективности генетического алгоритма, что связано с ростом сложности проблемы.

- Оптимальные значения вероятности мутации особи существенно варьируются в зависимости от размера графа и плотности ребер.

- Анализ результатов показал, что при увеличении количества вершин в графе время поиска наименьшего вершинного покрытия возрастает, однако алгоритм сохраняет свою эффективность для графов небольших размеров.

## **Заключение**

Экспериментальные исследования, проведенные на графах с различным количеством вершин, вероятностью существования ребер и вероятностью мутации особи, продемонстрировали эффективность подхода генетических алгоритмов к решению задач о вершинном покрытии графа.

Данная работа вносит вклад в изучение методов оптимизации для решения задачи нахождения наименьшего вершинного покрытия, которые могут быть использованы в различных областях, таких как сетевой анализ и комбинаторная оптимизация. Дальнейшие исследования могут быть сосредоточены на разработке более эффективных алгоритмов, оптимизации параметров генетического алгоритма для конкретных типов графов и расширении подхода на более сложные графовые структуры.



## Список литературы

1. Goldberg D.E. - “Genetic Algorithms in Search, Optimization and Machine Learning”, 2001
2. Goldberg D. E. A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing // Complex Syst. —Vol. 4. — P. 445–460.
3. Javier A Hassan, Andrea Maria, Rasid Regimar, Ami-Shab S. Sappari - “Certified Vertex Cover of a Graph”, 2024
4. Jinhui Duan - “The Minimum Vertex Cover Problem for a Specifically Constructed Planar 3-Regular Graph”, 2024
5. Holland J. Adaptation in natural and artificial systems. — Ann Arbor: Univ. Michigan Press, 1975. — 183 p.
6. Kaustav Paul, Atri Pandey - “Algorithms and Discrete Applied Mathematics” Eternal Connected Vertex Cover Problem in Graphs: Complexity and Algorithms
7. Neeldhara Misra, Saraswati Nanoti - “Eternal Vertex Cover on Bipartite and Co-Bipartite Graphs”, 2022
8. Sangram K. Jena, K. Subramani - “Theory and Applications of Models of Computation” Analyzing the 3-Path Vertex Cover Problem in Planar Bipartite Graphs, Jan 2023
9. Wang Luzhi, Chi-Min Li, Junping Zhou, Bo Jin, Minghao Yin - “An Extract Algorithm for the Minimum Weight Vertex Cover Problem in Large Graphs”
10. Береснев В., Мельников А., Утюпин С. - “Оптимизация и Приложения”, Представление Проблемы Покрытия Вечной Вершины как Динамической Игры Stackelberg, 2023
11. Батищев Д. И., Неймарк Е.А., Старостин Н.В. - “Применение Генетических Алгоритмов к Решению Задач Дискретной Оптимизации”
12. Гладков Л. А., Курейчик В. В., Курейчик В.М. Генетические Алгоритмы / Под ред. В.М. Курейчика. М.: ФИЗМАТЛИТ, 2010

13. Горбачевская Л.Е. Полиномиально Разрешимые и NP-Трудные Двухуровневые Задачи Стандартизации и Размещения”, 1998
14. Еремеев А.В. Генетический алгоритм с селекцией, 2011. – 13 с.
15. Еремеев А.В. Генетический алгоритм для задач о покрытии // Дискретный анализ и исследование операций, 2000. – Серия 2 Том 7, №16 – 47-60 с.
16. Листровой С.В., Мощный С.В.- “Алгоритм Решения Задачи о Наименьшем Вершинном Покрытии Произвольного Графа с Помощью Систем Квадратичных Уравнений”
17. Панченко Т. В. - “Генетические Алгоритмы”
18. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. — М: Горячая линия — Телеком, 2006. — 452 с.
19. Шиян В. И., Курченко Е. А., Мирошниченко А. Н. - сравнительный Анализ Жадного Алгоритма и Ленивого Алгоритма для Решения Задачи о Вершинном Покрытии Гиперграфа”

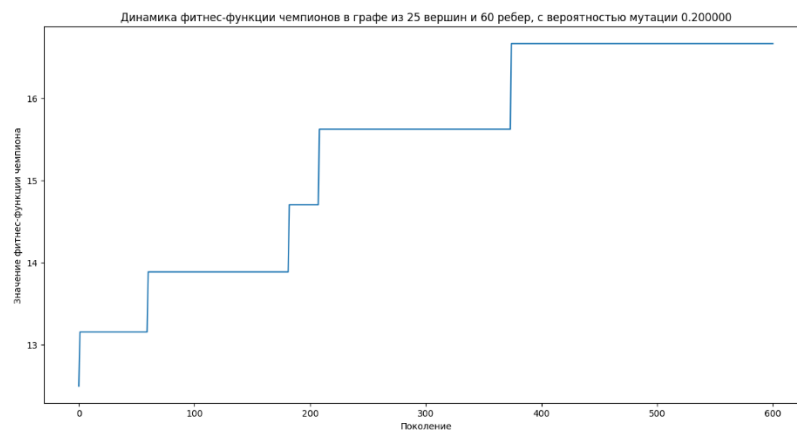
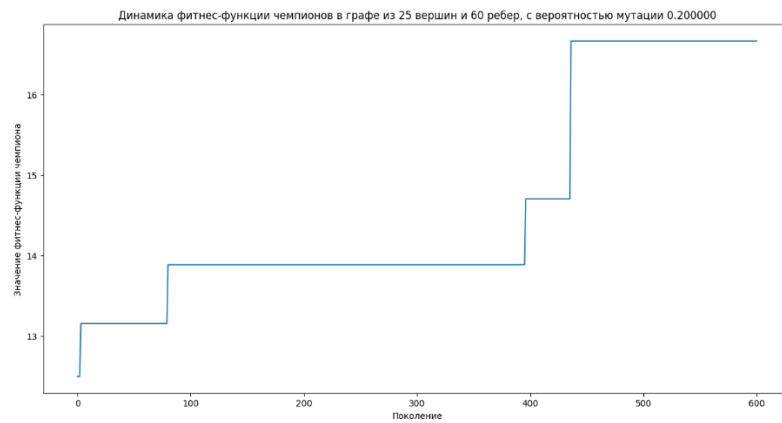
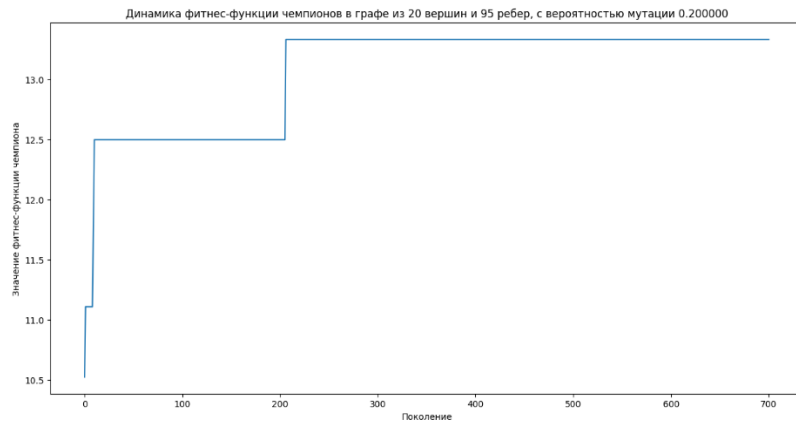
## Приложение 1

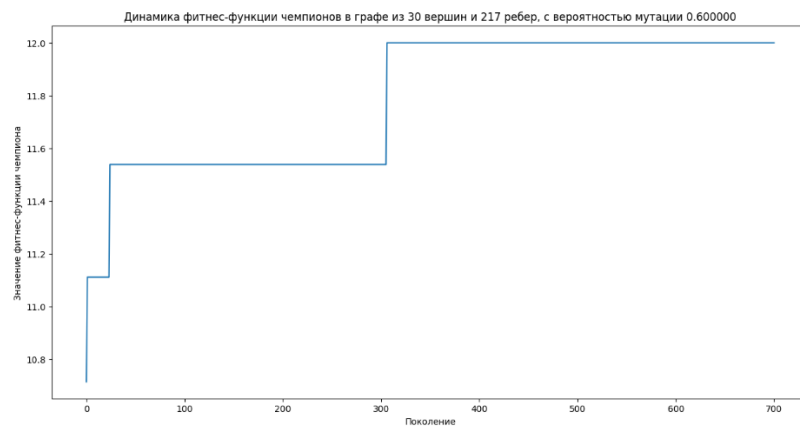
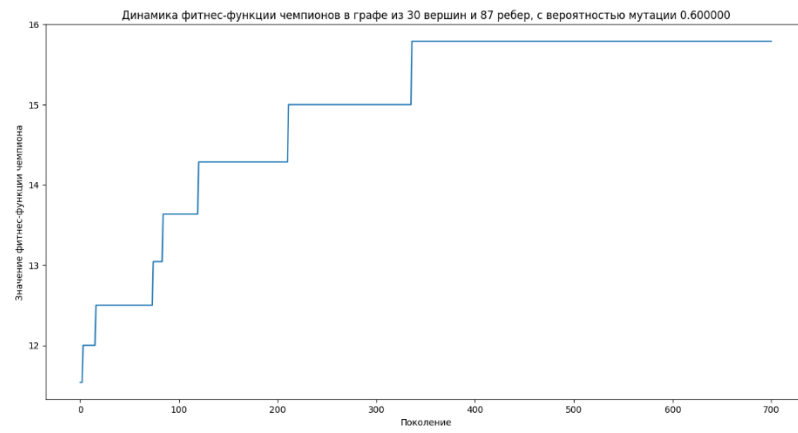
### Программная реализация.

Программная реализация генетического алгоритма находится по ссылке:  
<https://github.com/aellieme>

## Приложение 2

### Динамика функции приспособленности в различных экспериментах.





Группа графиков 2.