

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Кафедра «Информатика»
кафедра

ОТЧЕТ ОБ ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

Генетические алгоритмы
Тема

Преподаватель		В. В. Тынченко
	Подпись, дата	Инициалы, Фамилия
Студент	КИ19-17/1Б, №031939174	С. Е. Говорухина
	Номер группы, зачетной книжки	Подпись, дата
		Инициалы, Фамилия
Студент	КИ19-17/1Б, №031939174	А. К. Никитин
	Номер группы, зачетной книжки	Подпись, дата
		Инициалы, Фамилия

Красноярск 2022

1 Цель

Разработать генетический алгоритм (ГА), реализующий задачу раскраски графа.

2 Задачи

Разрабатываемая программа должна реализовывать следующие задачи:

- ручной ввод графа пользователем;
- нахождение хроматического числа графа;
- найденное хроматическое число графа должно быть минимальным или близким к минимальному;
- визуализация раскрашенного графа;
- возможность настройки параметров алгоритма;
- визуализация процесса обучения ГА;
- исследование влияния гиперпараметров на точность решения поставленной задачи.

3 Описание генетического алгоритма

3.1 Описание особи

3.1.1 Генотип

Пусть n – количество вершин графа.

Генотип особи представляет из себя массив целых чисел длины n , состоящий из чисел в диапазоне $[0, n - 1]$.

Каждое число соответствует закрепленным за ним цвету вершины графа, соответствующей позиции числа.

Пример генотипа – $[17, 12, 5, 5, 5, 12, 5, 17, 5, 12, 5, 12, 17, 17, 17, 17, 17, 5, 12, 12]$.

3.1.2 Фенотип

Фенотип особи является раскрашенным графом. Пример фенотипа указан на рисунке 1.

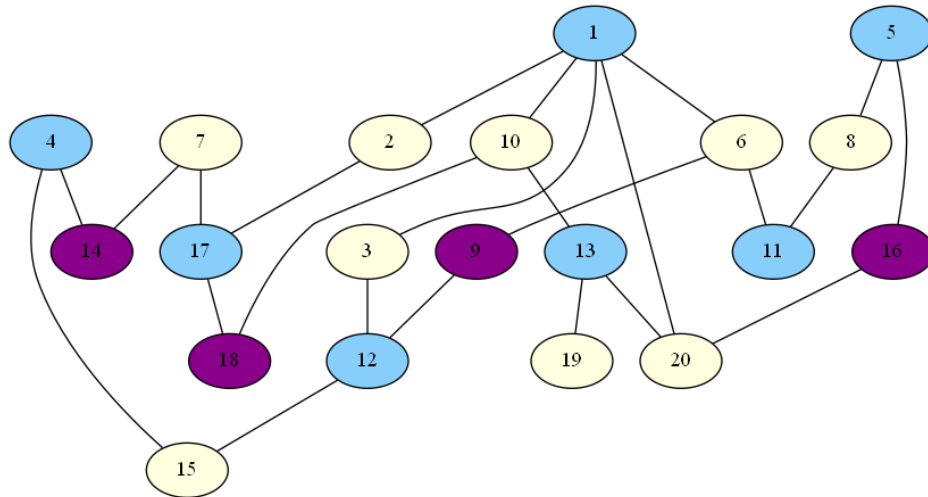


Рисунок 1 – Фенотип особи

3.2 Начальная популяция

Пусть n – количество вершин графа.

Особь начальной популяции представляет из себя массив длиной n , каждая ячейка которого является целым числом в промежутке $[0, n - 1]$, выбираемым случайным образом.

3.3 Оценка пригодности особи

Цель функции пригодности – оценить качество поданной на вход особи относительно других особей.

Было выделено два критерия сравнения особей.

1. Количество нарушений в раскраске графа. Данный критерий подсчитывает количество пар с одинаковыми смежными цветами.

2. Количество цветов. Данный критерий подсчитывает, сколько цветов имеет поданный на вход генотип.

Вклад каждого из критерия сравнения особи рассчитывается, исходя из заранее предопределенных параметров `HARD_CONSTRAINT_PENALTY` и `COLORS_AMOUNT_PENALTY`.

Таким образом, финальная оценка функции пригодности особей рассчитывается следующим образом:

$$final_score = HARD_CONSTRAINT_PENALTY * \text{количество_нарушений} + COLORS_AMOUNT_PENALTY.$$

3.4 Метод селекции особей

Селекция – это выбор тех хромосом, которые будут участвовать в создании потомков для следующей популяции, т.е. для очередного поколения. Такой выбор производится согласно принципу естественного отбора, по которому наибольшие шансы на участие в создании новых особей имеют хромосомы с наибольшими значениями функции приспособленности.

В качестве метода селекции была использована турнирная селекция.

При ней все особи популяции разбиваются на подгруппы по 2 с последующим выбором в каждой из них особи с наилучшей приспособленностью.

3.5 Метод скрещивания особей

Скрещивание (также называется кроссинговер и кроссовер) – основная базовая операция в генетическом алгоритме. Здесь перебираются пары родителей из отобранной популяции и с некоторой высокой вероятностью выполняется обмен фрагментами генетической информации для формирования хромосом двух потомков.

В качестве метода скрещивания было использовано двухточечное скрещивание.

Двухточечное скрещивание во многом походит на одноточечное. Случайным образом определяются две точки разреза хромосомы, и затем

соответствующие части меняются местами. Получаются две новые хромосомы для двух потомков.

Принцип работы двухточечного скрещивания демонстрируется на рисунке 2.

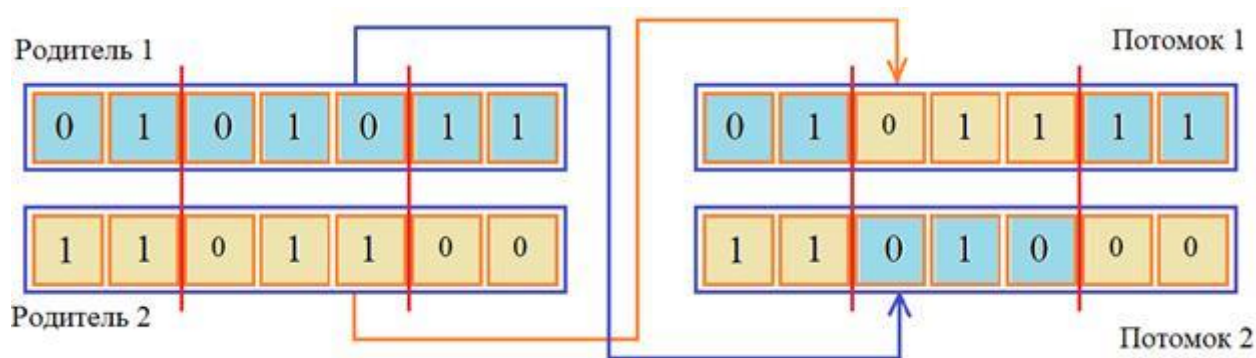


Рисунок 2 – Двухточечное скрещивание

3.6 Метод мутации особей

Мутация — генетический оператор, который с некой вероятностью меняет один или несколько генов в случайных позициях хромосомы.

Мутация, как и скрещивание, генерирует новую особь, однако, в отличие от скрещивания, это изменение происходит случайным образом. Это необходимо, чтобы предотвратить стагнацию и уход в локальные минимумы из-за неизменной информации родителей.

Был разработан свой собственный алгоритм мутации на основе равномерной мутации.

У каждой особи есть вероятность подвергнуться мутации. Если особь подверглась мутации, каждый из ее генов заменяется с некоторой вероятностью. Значение, на которое заменяется ген, зависит от лучшего решения предыдущего поколения и представляет из себя случайно взятый индекс цвета.

Данное решение позволило алгоритму мутации не использовать цвета, которые уже не используются в особях, что лишь ухудшало средний показатель популяции.

4 Результаты

Разработка алгоритма производилась на языке программирования Python с использованием библиотеки `dear` для ГА, `pandas` для работы с данными и `graphviz` для визуализации графа.

В качестве графа для демонстрации была использована модель, представленная на рисунке 3.

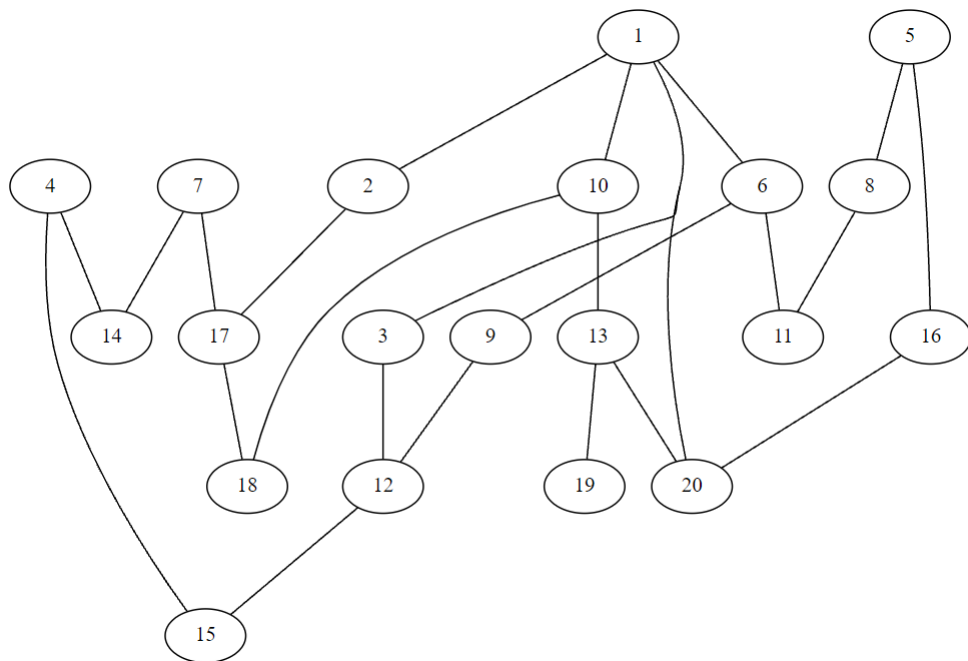


Рисунок 3 – Нераскрашенный граф

Было заранее рассчитано хроматическое число графа – 3 цвета. Затем были определены гиперпараметры для раскраски с помощью ГА.

Листинг 1 – Значения параметров ГА

```
POPULATION_SIZE = 100    # количество индивидуумов в популяции
P_CROSSOVER = 0.7        # вероятность скрещивания
P_MUTATION = 0.5         # вероятность мутации индивидуума
MAX_GENERATIONS = 120    # максимальное количество поколений
HALL_OF_FAME_SIZE = 1     # размер зала славы

HARD_CONSTRAINT_PENALTY = 10
COLOR_AMOUNT_PENALTY = 1
```

Полученный в результате работы ГА граф представлен на рисунке 4.

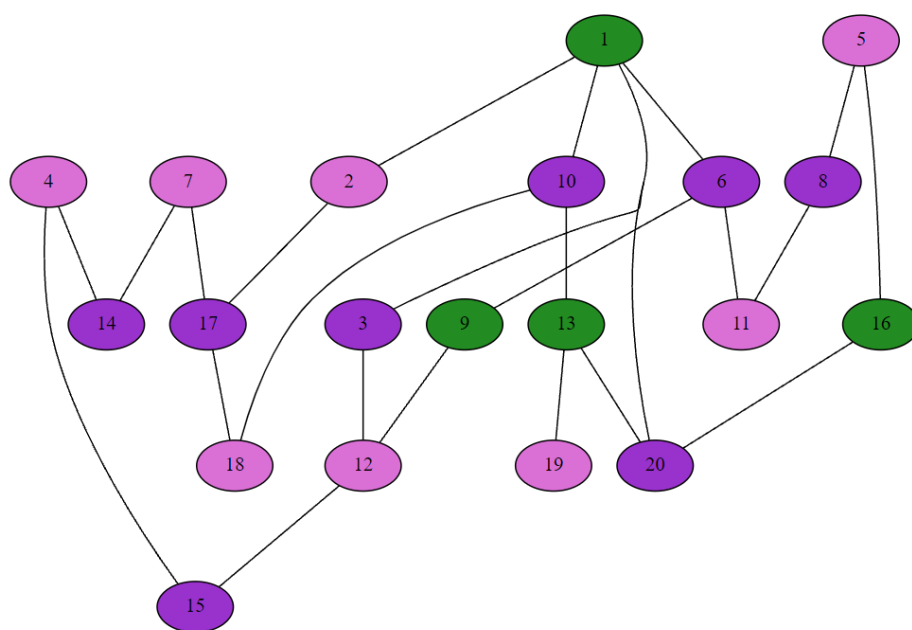


Рисунок 4 – Раскрашенный граф

Хроматическое число графа равно трем, что соответствует действительности.

Процесс обучения ГА наглядно представлен на рисунке 5.

Зависимость минимальной и средней приспособленности от поколения

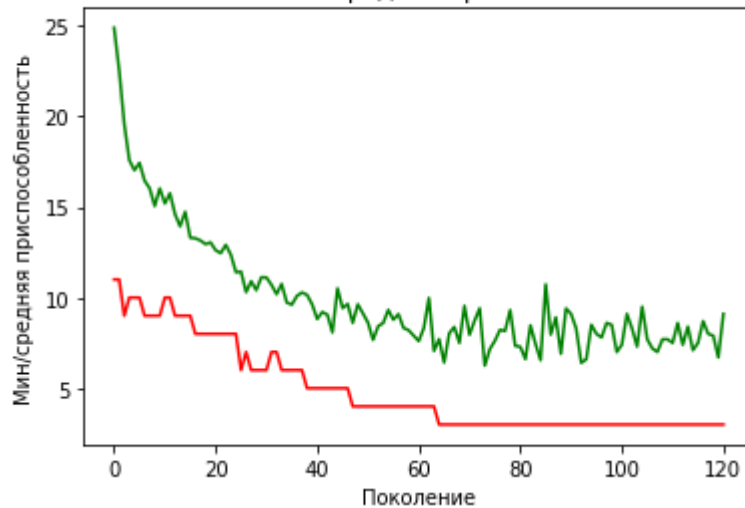


Рисунок 5 – Процесс обучения ГА

5 Исследование влияния параметров ГА на точность решения

5.1 Описание гиперпараметров

1. POPULATION_SIZE – количество особей в популяции.
2. P_CROSSOVER – вероятность скрещивания особей.
3. P_MUTATION – вероятность мутации особей.
4. MAX_GENERATIONS – максимальное количество поколений.
5. HARD_CONSTRAINT_PENALTY – коэффициент штрафа за нарушение раскраски графа.
6. COLOR_AMOUNT_PENALTY – коэффициент штрафа за количество цветов в генотипе особи.

5.2 Исследование гиперпараметров

В качестве моделей для сравнения гиперпараметров ГА было выбрано два графа. Первый граф из 20 вершин был представлен на рисунке 2, второй граф – граф Мычельского 6 порядка из 47 вершин.

Таблица с результатами исследования первого графа представлена на рисунке 6.

	population_size	p_crossover	p_mutation	hard_constraint_penalty	min_colors	min_generations
0	60.0	0.5	0.1	10.0	5.0	57.0
1	60.0	0.5	0.5	10.0	4.0	73.0
2	60.0	0.5	0.7	10.0	4.0	65.0
3	60.0	0.7	0.1	10.0	5.0	46.0
4	60.0	0.7	0.5	10.0	4.0	69.0
5	60.0	0.7	0.7	10.0	4.0	71.0
6	60.0	0.9	0.1	10.0	4.0	135.0
7	60.0	0.9	0.5	10.0	3.0	74.0
8	60.0	0.9	0.7	10.0	3.0	105.0
9	100.0	0.5	0.1	10.0	5.0	54.0
10	100.0	0.5	0.5	10.0	3.0	111.0
11	100.0	0.5	0.7	10.0	3.0	118.0
12	100.0	0.7	0.1	10.0	4.0	51.0
13	100.0	0.7	0.5	10.0	3.0	64.0
14	100.0	0.7	0.7	10.0	3.0	149.0
15	100.0	0.9	0.1	10.0	4.0	80.0
16	100.0	0.9	0.5	10.0	3.0	70.0
17	100.0	0.9	0.7	10.0	4.0	74.0
18	200.0	0.5	0.1	10.0	4.0	73.0
19	200.0	0.5	0.5	10.0	3.0	70.0
20	200.0	0.5	0.7	10.0	3.0	121.0
21	200.0	0.7	0.1	10.0	4.0	48.0
22	200.0	0.7	0.5	10.0	3.0	140.0
23	200.0	0.7	0.7	10.0	3.0	87.0
24	200.0	0.9	0.1	10.0	3.0	77.0
25	200.0	0.9	0.5	10.0	3.0	66.0
26	200.0	0.9	0.7	10.0	3.0	114.0
27	500.0	0.5	0.1	10.0	4.0	74.0
28	500.0	0.5	0.5	10.0	4.0	53.0
29	500.0	0.5	0.7	10.0	3.0	82.0
30	500.0	0.7	0.1	10.0	4.0	45.0
31	500.0	0.7	0.5	10.0	3.0	96.0
32	500.0	0.7	0.7	10.0	3.0	115.0
33	500.0	0.9	0.1	10.0	4.0	55.0
34	500.0	0.9	0.5	10.0	3.0	99.0
35	500.0	0.9	0.7	10.0	3.0	102.0

Рисунок 6 – Подбор гиперпараметров первого графа

Хроматическое число графа – 3, что удовлетворяет полученным результатам.

Исходя из результатов, можно предположить, что параметры мутации и размера популяции играют здесь ключевую роль и не должны быть слишком низкими.

Таблица с результатами исследования второго графа представлена на рисунке 7.

	population_size	p_crossover	p_mutation	hard_constraint_penalty	min_colors	min_generations
31	500.0	0.7	0.5	10.0	8.0	123.0
35	500.0	0.9	0.7	10.0	8.0	149.0
32	500.0	0.7	0.7	10.0	9.0	127.0
28	500.0	0.5	0.5	10.0	9.0	133.0
26	200.0	0.9	0.7	10.0	9.0	147.0
33	500.0	0.9	0.1	10.0	10.0	123.0
5	60.0	0.7	0.7	10.0	10.0	129.0
25	200.0	0.9	0.5	10.0	10.0	129.0
10	100.0	0.5	0.5	10.0	10.0	130.0
23	200.0	0.7	0.7	10.0	10.0	134.0
19	200.0	0.5	0.5	10.0	10.0	136.0
22	200.0	0.7	0.5	10.0	10.0	145.0
30	500.0	0.7	0.1	10.0	10.0	150.0
4	60.0	0.7	0.5	10.0	11.0	96.0
20	200.0	0.5	0.7	10.0	11.0	120.0
17	100.0	0.9	0.7	10.0	11.0	125.0
13	100.0	0.7	0.5	10.0	11.0	129.0
14	100.0	0.7	0.7	10.0	11.0	129.0
8	60.0	0.9	0.7	10.0	11.0	135.0
11	100.0	0.5	0.7	10.0	11.0	135.0
34	500.0	0.9	0.5	10.0	11.0	135.0
29	500.0	0.5	0.7	10.0	11.0	145.0
15	100.0	0.9	0.1	10.0	12.0	98.0
16	100.0	0.9	0.5	10.0	12.0	101.0
18	200.0	0.5	0.1	10.0	12.0	105.0
27	500.0	0.5	0.1	10.0	12.0	112.0
7	60.0	0.9	0.5	10.0	12.0	117.0
24	200.0	0.9	0.1	10.0	12.0	123.0
1	60.0	0.5	0.5	10.0	12.0	131.0
2	60.0	0.5	0.7	10.0	12.0	150.0
6	60.0	0.9	0.1	10.0	13.0	103.0
21	200.0	0.7	0.1	10.0	13.0	140.0
12	100.0	0.7	0.1	10.0	14.0	67.0
9	100.0	0.5	0.1	10.0	14.0	100.0

Рисунок 7 – Подбор гиперпараметров второго графа

Аналогичные предыдущим выводу можно применить и к этому графу, однако в глаза бросается следующая проблема. Хроматическое числа второго графа – 6, в то время как лучшая особь смогла достичь результата лишь 8. Это наводит на мысль, что ГА является не лучшим выбором для раскраски графов с большим количеством вершин и связей между ними.

6 Выводы

В данной исследовательской работе был реализован алгоритм раскраски простого графа посредством генетического алгоритма. Был сделан вывод, что генетический алгоритм целесообразно применять в случае малых графов, однако в случае больших графов он может привести к неверным результатам. Алгоритмический подход в которых простой алгоритмический подход в данном случае является наиболее предпочтительным, ведь ГА не может гарантировать того, что найденное значение действительно будет минимальным.

Также были проанализированы гиперпараметры метода и их влияние на работу функции. Оттуда был сделан вывод, что параметры мутации и количества особей в популяции лучше не делать слишком небольшим.