

# Задача раскраски графа в контексте оптимизации расписания: программное решение

О. И. Хонина, к.и.н. А. В. Забродин

Петербургский государственный университет путей сообщения Императора Александра I

Санкт-Петербург, Россия

olga.khonina.0311@mail.ru, teach-case@yandex.ru

**Аннотация.** Статья посвящена анализу проблемы раскраски графа, актуальной в различных областях, включая составление учебных расписаний, планирование и управление ресурсами и многих других. В ходе исследования эта задача была адаптирована для оптимизации расписания учебных занятий. Основной целью исследования стало создание алгоритма и программного обеспечения для разработки пользовательского интерфейса расписания в учебных учреждениях. Разработанный софт может быть легко настроен для различных типов учебных заведений: от школ до университетов.

**Ключевые слова:** граф, расписание, алгоритм, жадный алгоритм, задача раскраски графа, визуальное представление, программа, Python.

## ВВЕДЕНИЕ

В последние десятилетия теория графов стремительно развивается, вместе с расширением ее применения в различных областях. Одной из классических задач этой теории является задача раскраски графа, где каждой вершине необходимо присвоить определенный цвет так, чтобы смежные вершины имели разные цвета, и при этом общее количество цветов было минимальным. Эта задача входит в список из 21 NP-трудных задач, сформулированных Ричардом Карпом в 1972 году, и по-прежнему остается вызовом для науки, так как нет полиномиального алгоритма для точного решения.

Теория планирования играет важную роль в обеспечении эффективного управления ресурсами, временем и задачами в различных областях. Она позволяет определять цели, разрабатывать стратегии и следить за выполнением задач. Задача раскраски графа может быть использована для оптимизации распределения ресурсов в различных процессах.

В статье внимание сосредоточено на практическом применении задачи раскраски графов в области планирования, особенно в контексте составления учебных расписаний, оптимизации транспортных маршрутов, времени работы оборудования и многих других аналогичных сценариев. Эта задача актуальна и требует эффективных алгоритмических решений.

Важным аспектом хорошего расписания является его визуальное представление. Четкий и привлекательный дизайн расписания упрощает восприятие информации, что особенно важно в быстро меняющейся и динамичной среде. Визуально привлекательное расписание способствует более эффективному использованию информации и снижает вероятность ошибок и недопонимания.

## ОСНОВНЫЕ ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Большинство определений теории графов даются по-разному в различных источниках, в основном, они демонстрируют разные подходы и не противоречат друг другу [1–4].

Графом, в общем случае, называется совокупность двух множеств:  $G = (V, E)$ , где  $V$  – множество вершин,  $E \subset V \times V$  – множество связей между ними (множество ребер) (рис. 1).

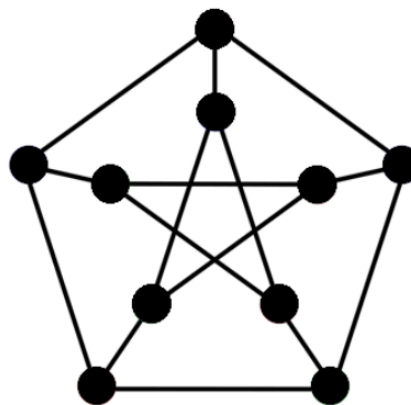


Рис. 1. Пример графа (граф Петерсена)

Две вершины  $u, v \in V$  называются смежными, если они связаны ребром:  $(u, v) \in E$ . Два ребра  $e, f \in E$  называются смежными, если они имеют общую вершину:  $e \cap f \neq \emptyset$ .

Степенью  $\deg(v)$  вершины  $v$  в графе  $G$  называют число вершин, смежных с вершиной  $v$ :  $\deg(v) = |\{u \in V \mid (v, u) \in E\}|$ .

Рассмотрим подробнее теорию раскраски графа [5].

Раскраской вершин графа  $G = (V, E)$  является функция  $c: V \rightarrow \mathbb{N}$ , которая каждой вершине графа ставит в соответствие натуральное число (цвет) так, что любым двум инцидентным вершинам  $u, v \in V$  назначаются разные цвета:  $\{u, v\} \in E \Rightarrow c(u) \neq c(v)$ . Функция  $c$  называется функцией раскраски (рис. 2).

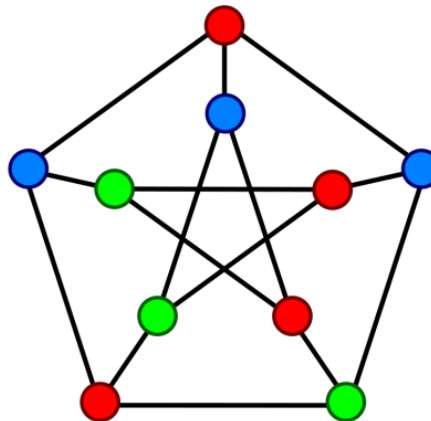


Рис. 2. Пример раскраски графа (граф Петерсена)

Применительно к статье функция раскраски имеет следующий вид:  $c: \{0, 1\}^{n \times n} \rightarrow \mathbb{N}n$ , то есть по матрице смежности графа мы получаем вектор его цветов.

Граф  $G$ , для которого существует раскраска из  $k$  цветов, называется  $k$ -раскрашиваемым. В этом случае функция раскраски разбивает граф  $G$  на независимые подмножества  $V_1, V_2, \dots, V_k$ , для которых справедливо:  $V_i \cap V_j = \emptyset$  и  $V_1 \cup V_2 \cup \dots \cup V_k = V$ .

Внутри независимого множества никакие две вершины не связаны ребром.

Наименьшее число  $k$ , для которого существует  $k$ -раскраска графа  $G$ , называется хроматическим числом графа  $G$  и обозначается  $\chi(G)$ . Любая раскраска графа  $G$ , для которой необходимо только  $k$  цветов и  $k = \chi(G)$ , называется оптимальной [6].

#### ОПИСАНИЕ ЗАДАЧИ

Задача раскраски графа представляет собой одну из ключевых в теории графов, так как она относится к классу NP-полных задач, и многие другие задачи этой области сводятся к ней. Раскраска графа может быть вершинной, реберной или тотальной; однако все эти варианты задачи взаимосвязаны, и в рамках нашей реализации ограничимся рассмотрением вершинной раскраски. При такой раскраске графа каждой вершине присваивается определенный цвет таким образом, что никакие две смежные вершины не имеют одного и того же цвета [7, 8]. Суть самой задачи состоит в нахождении минимального количества цветов для раскраски графа.

На основе вышесказанного задачу составления расписания можно интерпретировать как задачу раскраски графа следующим образом:

- преобразование задачи расписания в графовую модель: каждому событию сопоставляется вершина, а ребра между вершинами представляют конфликты времени (события, которые не могут произойти одновременно);
- раскраска графа таким образом, чтобы две смежные вершины (события, которые не могут происходить одновременно) всегда имели разные цвета; в этом контексте «цвет» представляет собой временной слот;
- минимизация количества используемых цветов, что соответствует минимизации общего числа временных слотов.

#### АЛГОРИТМЫ ДЛЯ РЕШЕНИЯ ЗАДАЧИ РАСКРАСКИ ГРАФА

Для решения задачи раскраски графа доступны различные методы, которые подразделяются на точные и приближенные [9, 10].

Точные алгоритмы обеспечивают нахождение оптимального решения, однако их сложность возрастает экспоненциально по мере увеличения размера графа. Сюда входят методы перебора, ветвления и границы, а также алгоритмы, основанные на теории множеств. Однако стоит учитывать, что для больших графов и графов с высоким хроматическим числом вычислительная сложность может быть существенной, и в этом случае стоит рассмотреть использование приближенных или эвристических методов.

Приближенные алгоритмы, в свою очередь, не обещают нахождение оптимального решения, но обладают высокой скоростью работы и могут быть применены к большим графам. К ним относятся жадные алгоритмы, генетические алгоритмы и другие подобные методы.

При решении рассматриваемой задачи возможно применение различных алгоритмов раскраски графа, однако конкретный выбор алгоритма зависит от размера графа, доступного времени и требуемой точности решения.

#### ПРИМЕНЕНИЕ АЛГОРИТМОВ РАСКРАСКИ ГРАФА ДЛЯ СОСТАВЛЕНИЯ РАСПИСАНИЯ

Для решения поставленной задачи будем использовать алгоритм жадной раскраски, который относится к приближенным методам.

Жадный алгоритм — это любой алгоритм, который следует эвристике решения проблемы, заключающейся в локально оптимальном выборе на каждом этапе. Во многих задачах жадная стратегия не дает оптимального решения, но жадная эвристика может дать локально оптимальные решения, которые приближаются к глобально оптимальному решению за разумный промежуток времени [11].

Жадный алгоритм для задачи раскраски графа имеет несколько преимуществ:

1. Алгоритм относительно прост в реализации и понимании. Он основан на интуитивно понятной идее выбора цвета для каждой вершины, начиная с первой и продолжая до последней.

2. Он работает быстро для большинства случаев и не требует выполнения сложных вычислений или итераций по всем вершинам графа. Вместо этого он работает локально, выбирая на каждом шаге наиболее подходящий доступный цвет для текущей вершины.

3. Жадный алгоритм обычно дает хорошие результаты раскраски графа и находит раскраску, которая использует не слишком много цветов и удовлетворяет большинству ограничений.

4. Данный алгоритм хорошо масштабируется и может быть применен к большим графам. Он не требует большого количества памяти или вычислительных ресурсов, поэтому может быть использован для решения задач раскраски графа с большим количеством вершин.

5. Жадный алгоритм может быть легко модифицирован или расширен для учета дополнительных ограничений или условий задачи раскраски графа. Это делает его удобным инструментом для решения различных вариантов задачи.

Однако следует отметить, что жадный алгоритм не всегда дает оптимальное решение и может быть подвержен ошибкам. В некоторых случаях он может привести к неправильной раскраске графа или использованию большего количества цветов, чем требуется.

Основные этапы жадного алгоритма:

1. Инициализация — присвоение всем вершинам «бесцветного» состояния.

2. Раскраска — выполнение следующих шагов для каждой вершины графа:

- а) выбор первой доступной вершины без цвета;
- б) присвоение вершине первого доступного цвета, который не используется ее соседями;
- в) повторение шагов а)–б) до тех пор, пока все вершины не получат цвет.

#### ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ

Разработанная программа, написанная на языке Python, предоставляет возможность чтения и визуализации расписания из файлов Excel (рис. 3), а также создания PDF-доку-

ментов с использованием скриншотов пользовательского интерфейса.

Понедель	1	Схем.	1-206	Гильвано	четная	2023-02-08
Вторник	2	Т.Вер.	7-307	Боровски	каждая	
Среда	3	САОД	1-206	Забродин	каждая	
Четверг	1	Схем.	4-305	Гильвано	каждая	
Пятница	3	БД	1-218	Молодки	четная	
Понедель	2	Инж.Гр.	2-314	Леонова	четная	
Понедель	3	Мат.	7-312	Рунев	каждая	
Понедель	4	Мат.	7-312	Рунев	нечетная	
Вторник	1	Т.Вер.	7-513	Боровски	четная	
Вторник	3	БД	1-217	Хомонен	каждая	
Понедель	4	Инж.Гр.	4-106	Леонова	четная	
Понедель	5	Т.Вер.	7-312	Боровски	нечетная	
Понедель	2	Фил.	4-312	Мурейко	нечетная	
Среда	2	Фил.	4-312	Мурейко	каждая	
Среда	4	Т.авт.	1-204	Гильвано	каждая	
Среда	5	Схем	1-204	Гильвано	нечетная	
Четверг	2	Т.авт.	4-305	Гильвано	каждая	
Четверг	3	Физ-ра	ДФК	Сизова	каждая	
Пятница	4	БД	1-218	Молодки	четная	

Рис. 3. Входные данные

За основу, как указывалось выше, был взят жадный алгоритм, который рассматривает занятия как вершины графа, а ограничения по времени как ребра, которые не могут иметь одинаковый цвет. В начале работы вершины сортируются по убыванию степени, и каждая вершина последовательно раскрашивается в наименьший доступный цвет, который еще не использован среди ее соседей.

Рассмотрим этапы применения алгоритма в данном контексте:

1. Цвета для ячеек выбираются из заранее заданного списка цветов (Colors).

2. Цвета циклически перебираются с помощью *itertools.cycle*. Таким образом, после использования последнего цвета из списка, алгоритм возвращается к первому цвету и начинает перебор снова.

3. Каждый учебный день недели отображается в отдельной колонке таблицы, для него алгоритм сохраняет предыдущий цвет, использованный для окрашивания ячейки с занятием.

4. При заполнении нового дня алгоритм выбирает следующий цвет из цикла. Если этот цвет уже использовался в предыдущем дне, алгоритм продолжает перебирать цвета из цикла до нахождения неиспользованного цвета.

5. Этапы 2–4 повторяются, пока все расписание не будет раскрашено.

Программа представляет собой графический интерфейс для отображения расписания, считываемого из файла Excel, и предоставляет возможность сохранения этого расписания в формате PDF.

#### ФУНКЦИИ ПРОГРАММЫ

Модель программы представлена на рисунке 4. Рассмотрим используемые в программе функции.

1. *Screenshot\_to\_pdf(window\_title, filename)*. Эта функция принимает два параметра: «window\_title» (заголовок окна, которое нужно «захватить» скриншотом) и «filename» (имя файла, в который нужно сохранить PDF).

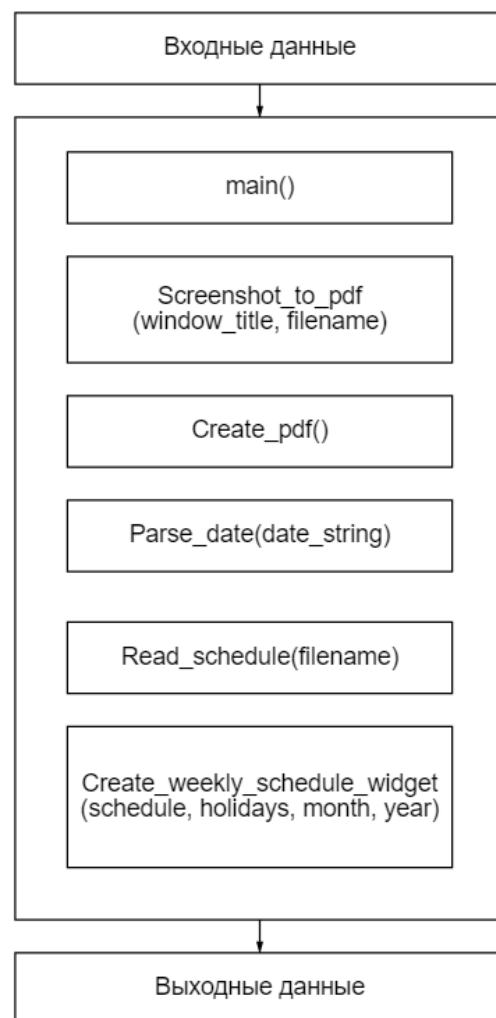


Рис. 4. Модель программы

Сначала функция использует библиотеку *pyautogui* для захвата скриншота окна. Затем скриншот обрезается для удаления оконных рамок. После этого функция создает PDF с использованием библиотеки FPDF и сохраняет его под указанным именем файла.

2. *Create\_pdf()*. Данная функция вызывает функцию *screenshot\_to\_pdf*, передавая ей следующие параметры: название окна «Schedule» и имя файла «output.pdf». Это приводит к созданию PDF-файла с изображением содержимого окна расписания.

3. *Parse\_date(date\_string)*. Функция принимает строку даты (*date\_string*) и преобразует ее в объект *date*, который более удобен для работы, кроме того, данная функция используется при чтении расписания из файла Excel.

4. *Read\_schedule(filename)*. Функция открывает файл Excel (*filename*) по указанному пути, считывает данные и сохраняет их в словарь. Возвращает два объекта: расписание и множество дней, считающихся исключениями (праздничные дни).

5. *Create\_weekly\_schedule\_widget(schedule, holidays, month, year)*. Функция создает виджет расписания для определенного месяца и года, используя предоставленное расписание и список выходных дней. Она воспроизводит распределение занятий по дням и времени с использованием

библиотеки *tkinter*. Каждая ячейка окрашивается в соответствующий цвет в зависимости от позиции и группы.

Код программы сначала загружает расписание из файла Excel, затем создает графический интерфейс с видом расписания и кнопкой для создания PDF. После этого программа запускает главный цикл обработки событий *tkinter*. Для реализации используются библиотеки: *tkinter* — для создания пользовательского интерфейса, *openpyxl* — для

чтения данных из Excel, *pyautogui* — для создания скриншотов и *FPDF* для создания PDF-файлов.

Реализованный в программе алгоритм обеспечивает уникальный цвет для каждого занятия в день и гарантирует, что цвета, используемые для раскраски занятий, изменяются от дня к дню. Это делает таблицу расписания более наглядной и легко читаемой.

Результат работы программы приведен на рисунке 5.

Февраль											
		Среда		Четверг		Пятница					
		1		2		3					
				Схем. (4-305, Гильванов)							
		Фил. (4-312, Мурейко)		Т.авт. (4-305, Гильванов)							
		САОД (1-206, Забродин)		Физ-ра (ДФК, Сизова)							
		Т.авт. (1-204, Гильванов)									
		Схем (1-204, Гильванов)									
		Понедельник		Вторник		Среда		Четверг		Пятница	
		6		7		8		9		10	
9:00-10:30		Схем. (1-206, Гильванов)				Выходной		Схем. (4-305, Гильванов)			
10:45-12:15		Инж.Гр. (2-314, Леонова)		Т.Вер. (7-307, Боровских)				Т.авт. (4-305, Гильванов)			
13:15-14:45		Мат. (7-312, Рунев)		БД (1-217, Хомоненко)				Физ-ра (ДФК, Сизова)		БД (1-218, Молодкин)	
15:00-16:30		Инж.Гр. (4-106, Леонова)								БД (1-218, Молодкин)	
16:45-18:15											
		Понедельник		Вторник		Среда		Четверг		Пятница	
		13		14		15		16		17	
9:00-10:30				Т.Вер. (7-513, Боровских)				Схем. (4-305, Гильванов)			
10:45-12:15		Фил. (4-312, Мурейко)		Т.Вер. (7-307, Боровских)		Фил. (4-312, Мурейко)		Т.авт. (4-305, Гильванов)			
13:15-14:45		Мат. (7-312, Рунев)		БД (1-217, Хомоненко)		САОД (1-206, Забродин)		Физ-ра (ДФК, Сизова)			
15:00-16:30		Мат. (7-312, Рунев)				Т.авт. (1-204, Гильванов)					
16:45-18:15		Т.Вер. (7-312, Боровских)				Схем (1-204, Гильванов)					
		Понедельник		Вторник		Среда		Четверг		Пятница	
		20		21		22		23		24	
9:00-10:30		Схем. (1-206, Гильванов)						Схем. (4-305, Гильванов)			
10:45-12:15		Инж.Гр. (2-314, Леонова)		Т.Вер. (7-307, Боровских)		Фил. (4-312, Мурейко)		Т.авт. (4-305, Гильванов)			
13:15-14:45		Мат. (7-312, Рунев)		БД (1-217, Хомоненко)		САОД (1-206, Забродин)		Физ-ра (ДФК, Сизова)		БД (1-218, Молодкин)	
15:00-16:30		Инж.Гр. (4-106, Леонова)				Т.авт. (1-204, Гильванов)				БД (1-218, Молодкин)	
16:45-18:15											
		Понедельник		Вторник							
		27		28							
9:00-10:30				Т.Вер. (7-513, Боровских)							
10:45-12:15		Фил. (4-312, Мурейко)		Т.Вер. (7-307, Боровских)							
13:15-14:45		Мат. (7-312, Рунев)		БД (1-217, Хомоненко)							
15:00-16:30		Мат. (7-312, Рунев)									
16:45-18:15		Т.Вер. (7-312, Боровских)									

Рис. 5. Результат работы программы

#### ЗАКЛЮЧЕНИЕ

Построение расписаний стало важной частью нашей повседневной жизни. Для ускорения этого процесса требуются эффективные решения, использующие современные технологии.

В статье представлен метод составления не только эффективного, но и визуально привлекательного расписания. Правильно структурированное расписание не только оптимизирует рабочие процессы, но также способствует снижению уровня стресса у персонала. Приоритетное распределение задач и ясная организация информации упрощают

восприятие расписания, делая его более доступным и понятным. Это, в свою очередь, способствует повышению мотивации и общей производительности.

Несомненно, что теория графов предоставляет мощные инструменты для оптимизации расписания, позволяя значительно уменьшить время, необходимое для нахождения оптимального решения. Однако даже с использованием современных методов и алгоритмов задача составления расписания по-прежнему остается сложной и многогранной, требующей дальнейших исследований.

Представленная программа успешно решает проблему раскраски расписания, что делает ее идеальным инструментом для создания расписания в образовательных учреждениях. Путем внесения определенных усовершенствований в код (добавление функций) ее потенциал можно значительно расширить и адаптировать под различные специфические требования.

В будущем можно улучшить данный метод, внедрив более эффективные алгоритмы раскраски графа и адаптируя его для решения специфических задач, таких как управление многозадачностью и учет многокритериальных параметров.

#### ЛИТЕРАТУРА

1. Харари, Ф. Теория графов = Graph theory / Пер. с англ. В. П. Козырева; под ред. Г. П. Гаврилова. — Москва: Мир. Редакция литературы по математическим наукам, 1973. — 300 с.
2. Оре, О. Графы и их применение = Graphs and their uses / Пер. с англ. Л. И. Головиной; под ред. И. М. Яглома. — Москва: Мир. Редакция литературы по математическим наукам, 1965. — 174 с. — (Современная математика).
3. Оре, О. Теория графов = Theory of graphs / Пер. с англ. И. Н. Врублевской; под ред. Н. Н. Воробьева. — 2-е изд., стер. — Москва: Наука. Главная редакция физико-математической литературы, 1980. — 336 с.
4. Берж, К. Теория графов и ее применения = Théorie des graphes et ses applications / Пер. с фр. А. А. Зыкова; под ред. И. А. Вайнштейна. — Москва: Изд-во иностранной литературы. Редакция литературы по математическим наукам, 1962. — 319 с.
5. Карпов, Д. В. Теория графов. Глава 5. Раскраски. — 2022. — 53 с. URL: [http://logic.pdmi.ras.ru/~dvk/ITMO/DM/2022-23/Лекции/5\\_colorings.pdf](http://logic.pdmi.ras.ru/~dvk/ITMO/DM/2022-23/Лекции/5_colorings.pdf) (дата обращения 16.09.2023).
6. Белов, В. В. Теория графов: Учебное пособие для втузов / В. В. Белов, Е. М. Воробьев, В. Е. Шаталов. — Москва: Высшая школа, 1976. — 392 с.
7. Татт, У. Т. Теория графов = Graph theory / Пер. с англ. Г. П. Гаврилова. — Москва: Мир. Редакция литературы по математическим наукам, 1988. — 424 с.
8. Раскраска графов // Википедия. — Обновлено 31.01.2023. URL: [http://ru.wikipedia.org/wiki/Раскраска\\_графов](http://ru.wikipedia.org/wiki/Раскраска_графов) (дата обращения 16.09.2023).
9. Родионов, В. В. Методы четырехцветной раскраски вершин плоских графов. — Москва: Ленанд, 2005. — 45 с.
10. Кристофидес, Н. Теория графов. Алгоритмический подход / пер. с англ. Э. В. Вершкова, И. В. Коновальцева; под ред. Г. П. Гаврилова. — Москва: Мир. Редакция литературы по математическим наукам, 1978. — 432 с.
11. Жадная раскраска // Википедия. — Обновлено 06.02.2021. URL: [http://ru.wikipedia.org/wiki/Жадная\\_раскраска](http://ru.wikipedia.org/wiki/Жадная_раскраска) (дата обращения 16.09.2023).

# Graph Coloring Problem in the Context of Schedule Optimization: Software Solution

O. I. Khonina, PhD A. V. Zabrodin

Emperor Alexander I St. Petersburg State Transport University

Saint Petersburg, Russia

olga.khonina.0311@mail.ru, teach-case@yandex.ru

**Abstract.** This article is dedicated to the analysis of the graph coloring problem, which is relevant in various fields, including scheduling, resource planning and management, among others. During the research, this problem was adapted for optimizing the scheduling of educational activities. The main goal of the study was to create an algorithm and software for developing a user interface for scheduling in educational institutions. The developed software can be easily customized for different types of educational establishments, from schools to universities.

**Keywords:** graph, schedule, algorithm, greedy algorithm, graph coloring problem, visual representation, program, Python.

## REFERENCES

1. Harary F. Graph Theory [Teoriya grafov]. Moscow, Mir Publishers, 1973, 300 p.
2. Ore O. Graphs and their uses [Grafy i ikh primeneniye]. Moscow, Mir Publishers, 1965, 174 p.
3. Ore O. Theory of graphs [Teoriya grafov]. Moscow, Nauka Publishers, 1980, 336 p.
4. Berge C. Théorie des graphes et ses applications [Teoriya grafov i ee primeneniya]. Moscow, Foreign Literature Publishing House, 1962, 319 p.
5. Karpov D. V. Graph Theory. Chapter 5. Colorings [Teoriya grafov. Glava 5. Raskraski], 2022, 53 p. Available at: [http://logic.pdmi.ras.ru/~dvk/ITMO/DM/2022-23/Лекции/5\\_colorings.pdf](http://logic.pdmi.ras.ru/~dvk/ITMO/DM/2022-23/Лекции/5_colorings.pdf) (accessed 16 Sep 2023).

6. Belov V. V., Vorobiev E. M., Shatalov V. E. Graph theory: Study guide for universities [Teoriya grafov: Uchebnoe posobie dlya vtuzov]. Moscow, Vysshaya Shkola Publishing House, 1976, 392 p.

7. Tutte W. T. Graph theory [Teoriya grafov]. Moscow, Mir Publishers, 1988, 424 p.

8. Graph Coloring [Raskraska grafov], *Wikipedia [Wikipedia]*. Last update at January 31, 2023. Available at: [http://ru.wikipedia.org/wiki/Раскраска\\_графов](http://ru.wikipedia.org/wiki/Раскраска_графов) (accessed 16 Sep 2023).

9. Rodionov V. V. Methods of four-coloring the vertices of planar graphs [Metody chetyrekhtsvetnoy raskraski vershin ploskikh grafov]. Moscow, Lenand, 2005, 45 p.

10. Christofides N. Graph theory. Algorithmic Approach [Teoriya grafov. Algoritmicheskiy podkhod]. Moscow, Mir Publishers, 1978, 432 p.

11. Greedy Coloring [Zhadnaya raskraska], *Wikipedia [Wikipedia]*. Last update at February 06, 2021. Available at: [http://ru.wikipedia.org/wiki/Жадная\\_раскраска](http://ru.wikipedia.org/wiki/Жадная_раскраска) (accessed 16 Sep 2023).