

Two-Step Colouring of Grid Graphs of Different Types

A. V. Smirnov¹DOI: [10.18255/1818-1015-2022-3-166-180](https://doi.org/10.18255/1818-1015-2022-3-166-180)¹P. G. Demidov Yaroslavl State University, 14 Sovetskaya str., Yaroslavl 150003, Russia.

MSC2020: 05C15

Research article

Full text in Russian

Received July 18, 2022

After revision August 30, 2022

Accepted September 2, 2022

In this article, we consider the NP-hard problem of the two-step colouring of a graph. It is required to colour the graph in a given number of colours in a way, when no pair of vertices has the same colour, if these vertices are at a distance of 1 or 2 between each other. The optimum two-step colouring is one that uses the minimum possible number of colours.

The two-step colouring problem is studied in application to grid graphs. We consider four types of grids: triangular, square, hexagonal, and octagonal. We show that the optimum two-step colouring of hexagonal and octagonal grid graphs requires 4 colours in the general case. We formulate the polynomial algorithms for such a colouring. A square grid graph with the maximum vertex degree equal to 3 requires 4 or 5 colours for a two-step colouring. In the paper, we suggest the backtracking algorithm for this case. Also, we present the algorithm, which works in linear time relative to the number of vertices, for the two-step colouring in 7 colours of a triangular grid graph and show that this colouring is always correct. If the maximum vertex degree equals 6, the solution is optimum.

Keywords: two-step graph colouring; grid graph; triangular grid graph; square grid graph; hexagonal grid graph; octagonal grid graph

INFORMATION ABOUT THE AUTHORS

Alexander Valeryevich Smirnov | orcid.org/0000-0002-0980-2507. E-mail: alexander_sm@mail.ru
correspondence author | PhD, Associate Professor, Department of Theoretical Computer Science.

Funding: This work was supported by P. G. Demidov Yaroslavl State University Project № VIP-016.

For citation: A. V. Smirnov, “Two-Step Colouring of Grid Graphs of Different Types”, *Modeling and analysis of information systems*, vol. 29, no. 3, pp. 166-180, 2022.

Двухшаговая раскраска графов решетки различных типов

А. В. Смирнов¹

DOI: [10.18255/1818-1015-2022-3-166-180](https://doi.org/10.18255/1818-1015-2022-3-166-180)

¹Ярославский государственный университет им. П. Г. Демидова, ул. Советская, д. 14, г. Ярославль, 150003 Россия.

УДК 519.174.7

Получена 18 июля 2022 г.

Научная статья

После доработки 30 августа 2022 г.

Полный текст на русском языке

Принята к публикации 2 сентября 2022 г.

В данной статье рассматривается NP-трудная задача о двухшаговой раскраске графа. Она состоит в нахождении такой раскраски в заданное число цветов, при которой ни одна пара вершин на расстоянии 1 или 2 друг от друга не будет окрашена в одинаковый цвет. Оптимальной считается двухшаговая раскраска в минимально возможное количество цветов.

Задача о двухшаговой раскраске исследуется применительно к графам решетки. Рассмотрены четыре типа решеток: треугольная, квадратная, шестиугольная и восьмиугольная. Показано, что в общем случае для оптимальной двухшаговой раскраски графов шестиугольной и восьмиугольной решетки требуется 4 цвета, приводятся полиномиальные алгоритмы получения такой раскраски. Для графа квадратной решетки, в котором максимальная степень вершины равна 3, может потребоваться 4 или 5 цветов для двухшаговой раскраски. В данной работе предложен алгоритм поиска с возвратом для этого случая. Для графов треугольной решетки представлен линейный относительно количества вершин алгоритм двухшаговой раскраски в 7 цветов, показано, что раскраска будет всегда корректной. Если максимальная степень вершины равна 6, решение будет оптимальным.

Ключевые слова: двухшаговая раскраска графа; граф решетки; граф треугольной решетки; граф квадратной решетки; граф шестиугольной решетки; граф восьмиугольной решетки

ИНФОРМАЦИЯ ОБ АВТОРАХ

Александр Валерьевич Смирнов
автор для корреспонденции

orcid.org/0000-0002-0980-2507. E-mail: alexander_sm@mail.ru
канд. физ.-мат. наук, доцент, кафедра теоретической информатики.

Финансирование: Работа выполнена в рамках инициативной НИР ЯрГУ им. П. Г. Демидова № VIP-016.

Для цитирования: A. V. Smirnov, "Two-Step Colouring of Grid Graphs of Different Types", *Modeling and analysis of information systems*, vol. 29, no. 3, pp. 166-180, 2022.

Введение

Двухшаговая раскраска графа – это такая раскраска, при которой все вершины, находящиеся на расстоянии 1 или 2 друг от друга, должны быть окрашены в разные цвета.

Подобная задача возникает при распределении временных слотов (тайм-слотов) в сетях, использующих принцип множественного доступа с временным разделением (см. [1]). В частности, при таком распределении возникает проблема скрытых узлов. Два узла сети могут находиться в зоне видимости третьего узла, но быть при этом невидимыми друг другу. Если эти два узла получают одинаковый номер тайм-слота, они будут передавать данные одновременно, что приведет к проблемам на третьем узле. По сути, каждые три узла сети, для которых возникает описанная проблема, должны получить разные номера тайм-слотов, что в точности соответствует условию двухшаговой раскраски вершин графа.

Очевидно, что рассматриваемая задача обобщает классическую задачу о раскраске графа (см., например, [2]). Так же, как и классическая, задача о двухшаговой раскраске является NP-трудной (см. [3]), поэтому важно определить полиномиальные подклассы задачи. К таковым относятся, например, прямоугольные графы решетки (см. [4]), которые можно раскрасить в 5 цветов за линейное относительно количества вершин время.

В данной статье мы рассмотрим графы, построенные на решетках различных типов (треугольной, квадратной, шестиугольной и восьмиугольной), и получим алгоритмы двухшаговой раскраски для них. Алгоритмы оптимальной раскраски графов шестиугольной и восьмиугольной решетки, а также раскраски графов треугольной решетки в 7 цветов будут полиномиальными.

1. Постановка задачи

Рассмотрим произвольный связный неориентированный граф $G(V, E)$. Напомним несколько определений из [3].

Определение 1. Будем называть вершину $v_2 \in V$ соседом первого порядка для некоторой вершины $v_1 \in V$, если v_1 и v_2 – смежные вершины, то есть существует ребро $\{v_1, v_2\} \in E$. Множество всех соседей первого порядка для вершины $v \in V$ будем обозначать через N_v^1 .

Определение 2. Соседом второго порядка для $v_1 \in V$ назовем такую вершину $v_2 \in V$, что кратчайшее расстояние между v_1 и v_2 равно двум. Иначе говоря, v_1 и v_2 – соседи второго порядка, если $\{v_1, v_2\} \notin E$ и найдется такая вершина $v_3 \in V$, что $\{v_1, v_3\} \in E$ и $\{v_2, v_3\} \in E$. Множество соседей второго порядка для $v \in V$ обозначим через N_v^2 .

Определение 3. Множество соседей первого и второго порядка для некоторой вершины $v \in V$ будем называть соседями v и обозначать через $N_v = N_v^1 \cup N_v^2$.

Теперь дадим постановку задачи о двухшаговой раскраске произвольного графа.

Задача 1 (двухшаговая раскраска графа). Пусть заданы связный неориентированный граф $G(V, E)$ и натуральное число K – количество цветов. Требуется получить раскраску G в K цветов таким образом, чтобы для каждой вершины $v \in V$ ни один из ее соседей первого или второго порядка не был окрашен в тот же цвет.

Иными словами, требуется определить функцию раскраски $c : V \rightarrow \{0, \dots, K - 1\}$ так, что для любой вершины $v \in V$ и для любой вершины $v' \in N_v$ выполнено

$$c(v) \neq c(v').$$

Оптимальной считается двухшаговая раскраска в минимально возможное число цветов K .

2. Графы решетки

Графы решетки – это планарные графы, порождаемые правильными евклидовыми паркетам. Обычно под графом решетки понимают граф, заданный на целочисленной квадратной решетке, однако часто рассматривают также графы треугольной, шестиугольной и восьмиугольной решетки.

Напомним, что *паркет* – это такое покрытие плоскости правильными многоугольниками, при котором два многоугольника имеют либо общую сторону, либо общую вершину, либо вообще не имеют общих точек. *Правильный паркет* всегда можно наложить на самого себя так, что любая заданная вершина наложится на любую другую заданную вершину (см., например, [5]). Тогда бесконечный граф решетки можно определить следующим образом.

Определение 4. Бесконечный граф решетки $G^\infty(V^\infty, E^\infty)$ – это граф, все вершины которого лежат во всех вершинах многоугольников некоторого правильного паркета, а множество ребер образовано всеми сторонами этих многоугольников.

Таким образом, бесконечный граф треугольной решетки задается правильным паркетом, состоящим из равносторонних треугольников; квадратной решетки – из квадратов; шестиугольной решетки – из правильных шестиугольников; восьмиугольной решетки – из правильных восьмиугольников и квадратов. Фрагменты бесконечных графов решетки указанных видов приведены на рис. 1.

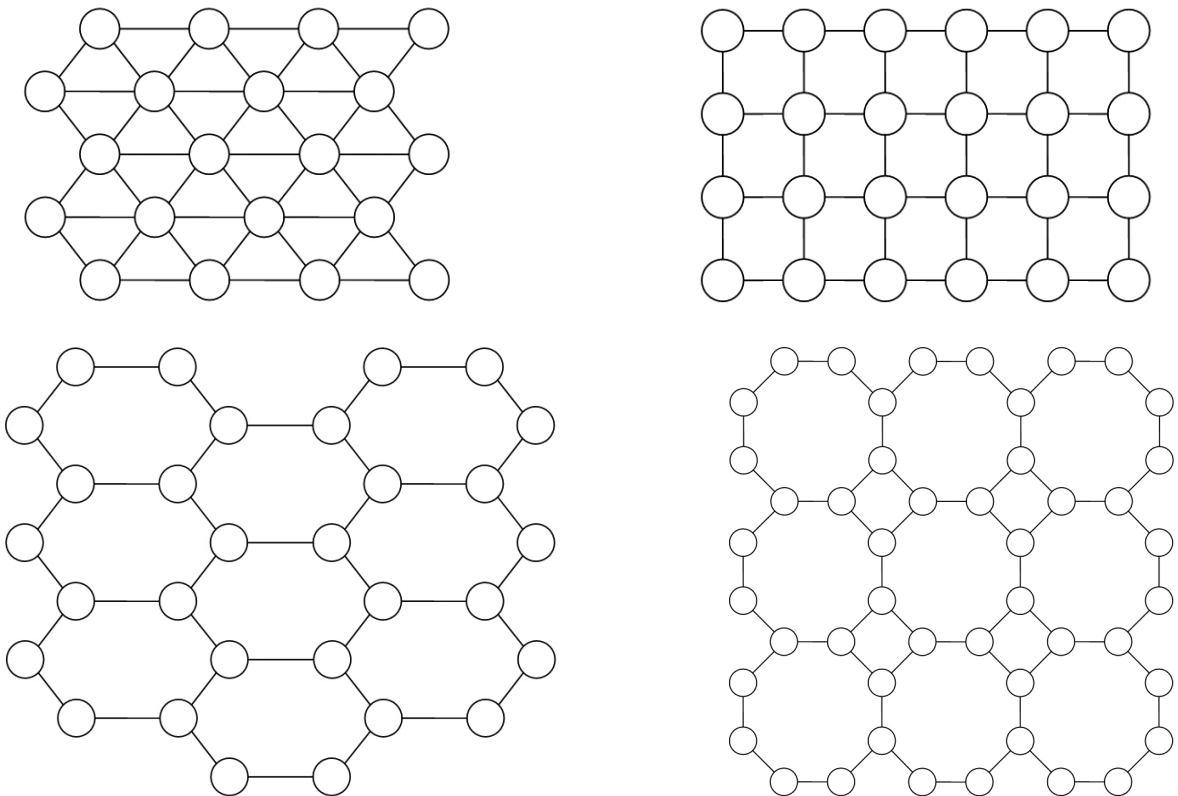


Fig. 1. Fragments of the infinite triangular, square, hexagonal, and octagonal grid graphs

Рис. 1. Фрагменты бесконечных графов треугольной, квадратной, шестиугольной и восьмиугольной решетки

Определение 5. Конечный граф решетки $G(V, E)$, или просто граф решетки – это конечный подграф бесконечного графа решетки $G^\infty(V^\infty, E^\infty)$, где множество E состоит из ребер, которые соединяют все вершины из V , находящиеся на расстоянии 1 друг от друга.

В данной статье мы будем рассматривать только связные графы решетки и строить двухшаговую раскраску для них (если граф несвязен, то каждая компонента связности раскрашивается независимо от остальных).

Граф решетки может быть *плотным* либо иметь *дыры*. В статье [6] плотный граф квадратной решетки (или граф без дыр) определен как граф решетки, все внутренние грани которого имеют площадь 1. Под гранями понимаются области, на которые граф решетки делит плоскость. Дадим аналогичное определение для графа произвольной решетки.

Определение 6. *Граф решетки $G(V, E)$ называется плотным, или графом без дыр, если каждая из его внутренних граней соответствует в точности одному многоугольнику со стороной длины 1, являющемуся элементом паркета, на основе которого построена решетка.*

Определение 7. *Граф решетки назовем вырожденным, если в нем не существует вершин степени большей 2.*

Очевидно, что вырожденный граф решетки – это либо только одна вершина, либо простая цепь, либо простой цикл.

Отметим, что, несмотря на простую структуру графов решетки, многие труднорешаемые задачи на графах остаются таковыми и для графов решетки. Так, например, в работах [7–9] доказана NP-полнота задачи о гамильтоновом цикле для графов квадратной, треугольной и шестиугольной решетки соответственно.

Что касается классической задачи о раскраске, для графов решетки рассматриваемых видов она становится тривиальной. Действительно, графы квадратной, шестиугольной и восьмиугольной решетки являются двудольными, поэтому всегда могут быть раскрашены в два цвета за линейное относительно количества вершин время, а граф треугольной решетки может быть раскрашен в три цвета также за линейное время. Однако задача о двухшаговой раскраске является более сложной нежели классическая, и даже для графов довольно простой структуры может потребоваться переборный алгоритм для нахождения оптимальной раскраски.

3. Двухшаговая раскраска цепей и циклов

Прежде чем говорить о двухшаговой раскраске графов решетки, рассмотрим правила раскраски графов, состоящих из одной простой цепи или из одного простого цикла, в минимально возможное количество цветов.

Правила двухшаговой раскраски простых цепей:

- цепь длины 0 (одна изолированная вершина) раскрашивается в 1 цвет;
- цепь длины 1 раскрашивается в 2 цвета;
- все остальные цепи раскрашиваются в 3 цвета в последовательности (0, 1, 2, 0, 1, 2, ...).

Правила двухшаговой раскраски простых циклов:

- цикл длины 5 раскрашивается в 5 цветов (все вершины являются соседями друг друга);
- цикл с числом вершин равным $3m$ ($m > 0$) раскрашивается в 3 цвета в последовательности (0, 1, 2, 0, 1, 2, ...);
- цикл с числом вершин равным $3m + 1$ ($m > 0$) раскрашивается в 4 цвета; для этого надо одну вершину окрасить в цвет 3, а для всех остальных использовать последовательность цветов (0, 1, 2, 0, 1, 2, ...);
- цикл с числом вершин равным $3m + 2$ ($m > 1$) раскрашивается в 4 цвета; для этого надо восемь вершин окрасить в 4 цвета в последовательности (0, 1, 3, 2, 0, 3, 1, 2), а для оставшихся $3(m - 2)$ вершин использовать последовательность цветов (0, 1, 2, 0, 1, 2, ...).

Последние два правила универсальны и не зависят от значения m , однако при конкретном m могут быть реализованы и другие варианты раскраски цикла.

Отметим, что мы рассмотрели все возможные варианты графов со степенью вершины не больше 2. В частности, пользуясь сформулированными правилами, можно за полиномиальное время осуществить двухшаговую раскраску любого вырожденного графа решетки в минимальное количество цветов. При этом вырожденный граф решетки не может представлять собой простой цикл длины 5. Действительно, все циклы на квадратной, шестиугольной или восьмиугольной решетке будут иметь четную длину ввиду двудольности графов на этих решетках, а любой граф треугольной решетки с числом вершин $|V| = 5$ представляет собой либо простую цепь, либо трапецию со сторонами длины 2, 1, 1, 1. В данной трапеции вершины верхнего основания находятся на расстоянии 1 от вершины, стоящей в середине нижнего основания, поэтому они будут соединяться с ней ребрами, то есть граф не будет вырожденным (есть вершины степени 3 и 4). Но в любом случае такая трапеция может быть раскрашена только в 5 цветов, поскольку все вершины являются соседями друг друга.

Сформулируем также полиномиальный алгоритм раскраски в минимальное количество цветов простой цепи, у которой фиксированы цвета четырех вершин: двух в начале и двух в конце цепи. Этот алгоритм мы будем использовать при получении двухшаговой раскраски графа квадратной решетки с ограниченной степенью вершин в разделе 5. Итак, пусть имеется простая цепь

$$P = (v_0, v_1, v_2, \dots, v_{n-1}, v_n), \quad n \geq 4,$$

где вершинам v_0, v_1, v_{n-1} и v_n присвоены цвета из множества $\{0, 1, 2, 3\}$. Цвета двух вершин могут совпадать, если это не противоречит правилам двухшаговой раскраски. Требуется получить двухшаговую раскраску цепи P в минимальное количество цветов при условии, что цвета вершин v_0, v_1, v_{n-1} и v_n менять нельзя. Естественно, это условие усложняет раскраску, и оптимальное количество цветов может оказаться больше 3.

В алгоритме мы будем использовать c_i ($i \in \overline{0, 4}$) для обозначения цвета вершин. При этом все значения $c_i \in \overline{0, 4}$ и $c_i \neq c_j$, если $i \neq j$.

Алгоритм 1.

1. Если $n = 4$, то надо окрасить единственную вершину v_2 . Все заранее окрашенные вершины v_0, v_1, v_3 и v_4 являются соседями v_2 . Поэтому для окраски v_2 используем цвет с минимальным номером, отличающийся от цветов соседей. В зависимости от цветов вершин v_0, v_1, v_3 и v_4 мы получим двухшаговую раскраску цепи P в 3, 4 или 5 цветов.

2. Если $n = 5$, то надо окрасить две вершины: v_2 и v_3 . Раскраску производим в зависимости от цветов вершин v_0, v_1, v_4 и v_5 в соответствии с таблицей (заранее окрашенные вершины отделены двойной чертой):

v_0	v_1	v_2	v_3	v_4	v_5	Оптимальная раскраска
c_0	c_1	c_2	c_3	c_0	c_1	4 цвета
c_0	c_1	c_2	c_3	c_1	c_0	4 цвета
c_0	c_1	c_2	c_3	c_0	c_2	4 цвета
c_0	c_1	c_3	c_4	c_2	c_0	5 цветов
c_0	c_1	c_2	c_0	c_1	c_2	3 цвета
c_0	c_1	c_3	c_0	c_2	c_1	4 цвета
c_0	c_1	c_3	c_0	c_2	c_3	4 цвета

3. Если $n = 6$, то надо окрасить три вершины: v_2, v_3 и v_4 . Раскраску производим в зависимости от цветов вершин v_0, v_1, v_5 и v_6 в соответствии с таблицей:

v_0	v_1	v_2	v_3	v_4	v_5	v_6	Оптимальная раскраска
c_0	c_1	c_2	c_4	c_3	c_0	c_1	5 цветов
c_0	c_1	c_2	c_0	c_3	c_1	c_0	4 цвета
c_0	c_1	c_2	c_3	c_1	c_0	c_2	4 цвета
c_0	c_1	c_2	c_0	c_1	c_2	c_0	3 цвета
c_0	c_1	c_2	c_0	c_3	c_1	c_2	4 цвета
c_0	c_1	c_2	c_0	c_3	c_2	c_1	4 цвета
c_0	c_1	c_2	c_0	c_1	c_2	c_3	4 цвета

4. Если $n = 7$, то надо окрасить четыре вершины: v_2, v_3, v_4 и v_5 . Раскраску производим в зависимости от цветов вершин v_0, v_1, v_6 и v_7 в соответствии с таблицей:

v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	Оптимальная раскраска
c_0	c_1	c_2	c_0	c_1	c_2	c_0	c_1	3 цвета
c_0	c_1	c_2	c_0	c_3	c_2	c_1	c_0	4 цвета
c_0	c_1	c_2	c_0	c_1	c_3	c_0	c_2	4 цвета
c_0	c_1	c_2	c_0	c_1	c_3	c_2	c_0	4 цвета
c_0	c_1	c_3	c_0	c_2	c_3	c_1	c_2	4 цвета
c_0	c_1	c_2	c_0	c_1	c_3	c_2	c_1	4 цвета
c_0	c_1	c_2	c_0	c_3	c_1	c_2	c_3	4 цвета

5. Если $n > 7$, то надо выполнить следующее. Пусть вершины v_0 и v_1 окрашены в цвета c_0 и c_1 соответственно. Если вершины v_{n-1} и v_n окрашены в те же цвета (в любом порядке), то выбираем в качестве c_2 цвет с минимальным номером, не совпадающий с c_0 и c_1 . Иначе c_2 – это цвет одной из вершин v_{n-1} и v_n , не совпадающий с c_0 и c_1 .

Цепь P можно окрасить в 3 цвета только в одном случае, когда при окрашивании вершин цепи, начиная с v_2 , в последовательности $(c_2, c_0, c_1, c_2, c_0, c_1, \dots)$ цвета вершин v_{n-1} и v_n совпадут с цветами последовательности. Фактически это означает, что при подобном окрашивании цепи цвета вершин v_{n-7} и v_{n-6} равны цветам вершин v_{n-1} и v_n соответственно (как в 1 строке таблицы шага 4). Во всех остальных случаях цепь P можно раскрасить в 4 цвета.

В любом случае, будем раскрашивать вершины с v_2 по v_{n-6} в последовательности $(c_2, c_0, c_1, c_2, c_0, c_1, \dots)$. После этого останутся неокрашенными вершины $v_{n-5}, v_{n-4}, v_{n-3}$ и v_{n-2} . Для их раскраски возьмем цепь $P' = (v_{n-7}, \dots, v_n) \subset P$ и применим к ней шаг 4 алгоритма.

4. Двухшаговая раскраска графов шестиугольной и восьмиугольной решетки

Графы шестиугольной и восьмиугольной решетки являются наиболее простыми представителями графов решетки в плане получения оптимальной двухшаговой раскраски. В данном разделе мы обоснуем полиномиальность задачи для данных классов графов. Сначала сформулируем соответствующие полиномиальные алгоритмы.

Пусть имеется произвольный невырожденный граф шестиугольной решетки $H(V, E)$. Найдем его двухшаговую раскраску с помощью следующего алгоритма.

Алгоритм 2 (двухшаговая раскраска графа шестиугольной решетки в 4 цвета).

1. Если граф $H(V, E)$ не является плотным, сделаем его таковым, добавив все возможные внутренние вершины и ребра. Если в графе есть ребра, не принадлежащие ни одной внутренней грани (это могут быть ребра цепи, соединяющей два плотных подграфа либо плотный подграф и висячую вершину), то построим на всех таких ребрах шестиугольные грани со стороной длины 1. Полученный граф обозначим через $H_S(V_S, E_S)$.

2. Граф $H_S(V_S, E_S)$ состоит из шестиугольных граней, располагающихся на разных вертикальных уровнях шестиугольной решетки, причем уровень определяется позицией самой левой вершины

шестиугольника. Присвоим всем граням графа номер вертикальной позиции от 0 до $m - 1$: грани, стоящие выше всех, получают номер 0, ниже всех – номер $m - 1$.

3. Все шестиугольные грани графа $H_S(V_S, E_S)$ также можно разделить на столбцы по позиции самой левой вершины грани, при этом в каждом столбце все грани имеют номера вертикальных позиций одинаковой четности и столбцы с четными/нечетными номерами чередуются. Пронумеруем все столбцы от 0 до n , если первый столбец содержит четные номера вертикальных позиций, и от 1 до n в противном случае.

4. Таким образом, каждая шестиугольная грань характеризуется парой координат (i, j) , где i – номер столбца, а j – номер вертикальной позиции. Выполним двухшаговую раскраску графа $H_S(V_S, E_S)$ по следующим правилам ($k \geq 0, l \geq 0$):

- грани с координатами $(4k, 4l)$ и $(4k + 2, 4l + 2)$ окрашиваются в последовательности $(0, 1, 2, 0, 1, 2)$;
- грани с координатами $(4k + 1, 4l + 1)$ и $(4k + 3, 4l + 3)$ окрашиваются в последовательности $(1, 0, 3, 1, 0, 3)$;
- грани с координатами $(4k, 4l + 2)$ и $(4k + 2, 4l)$ окрашиваются в последовательности $(3, 2, 1, 3, 2, 1)$;
- грани с координатами $(4k + 1, 4l + 3)$ и $(4k + 3, 4l + 1)$ окрашиваются в последовательности $(2, 3, 0, 2, 3, 0)$.

Окрашивание всех граней производится по часовой стрелке, начиная с самой левой вершины.

5. После этого накладываем раскраску графа $H_S(V_S, E_S)$ на граф $H(V, E)$ таким образом, чтобы общие вершины двух графов получили одинаковые цвета в каждом из них.

Пример раскраски графа шестиугольной решетки по алгоритму 2 приведен на рис. 2 (внутри граней подписаны их координаты, внутри вершин – номера цветов).

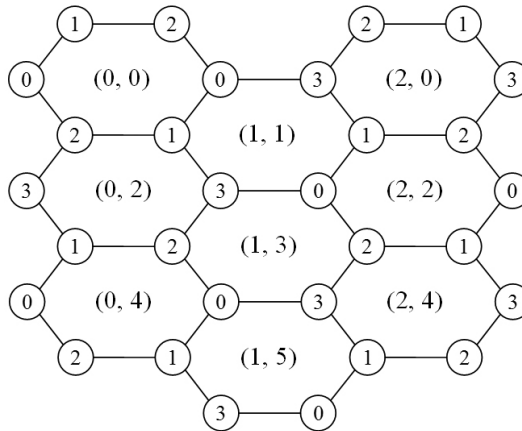


Fig. 2. Two-step colouring of the hexagonal grid graph

Рис. 2. Двухшаговая раскраска графа шестиугольной решетки

Пусть имеется произвольный невырожденный граф восьмиугольной решетки $O(V, E)$. Найдем его двухшаговую раскраску с помощью следующего алгоритма.

Алгоритм 3 (двухшаговая раскраска графа восьмиугольной решетки в 4 цвета).

Раскрасим все вершины графа $O(V, E)$, исходя из того соображения, что все квадратные грани бесконечного графа восьмиугольной решетки должны быть раскрашены в последовательности $(0, 1, 2, 3)$ по часовой стрелке, начиная с верхней вершины.

Отметим, что алгоритм 3 будет выполнять двухшаговую раскраску за линейное относительно количества вершин время. Пример раскраски графа восьмиугольной решетки по алгоритму 3 приведен на рис. 3.

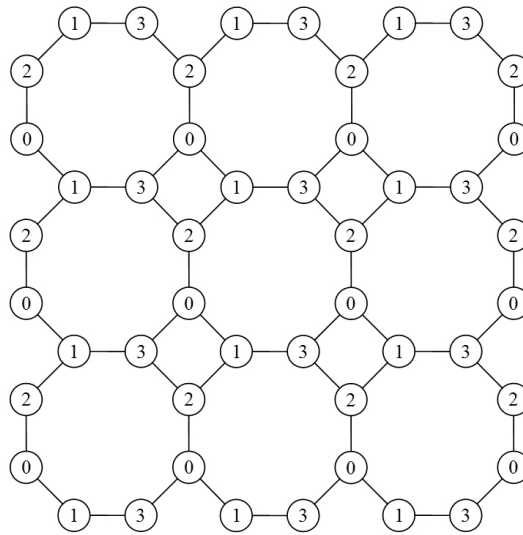


Fig. 3. Two-step colouring of the octagonal grid graph

Рис. 3. Двухшаговая раскраска графа восьмиугольной решетки

Теорема 1. *Оптимальная двухшаговая раскраска любого невырожденного графа шестиугольной или восьмиугольной решетки требует 4 цвета. Эта раскраска всегда может быть найдена с помощью полиномиальных алгоритмов 2 и 3.*

Доказательство. Сначала заметим, что для двухшаговой раскраски невырожденных графов шестиугольной и восьмиугольной решетки требуется не менее 4 цветов, поскольку максимальная степень вершины в таких графах равна 3 (свойство 1 задачи о двухшаговой раскраске графа, см. [3]). Поэтому для доказательства теоремы нужно показать, что раскраски в 4 цвета, определяемые алгоритмами 2 и 3, корректны.

Рассмотрим раскраску графа шестиугольной решетки $H_S(V_S, E_S)$ из алгоритма 2. Возьмем произвольную грань с координатами (i, j) . Заметим, что соседи каждой вершины этой грани могут находиться в текущей грани, а также в гранях, имеющих общее ребро с текущей. Внутри каждой грани раскраска корректна. Для каждой вершины рассматриваемой грани проверим цвета соседей из других граней. Для определенности будем считать, что мы рассматриваем грань с раскраской $(0, 1, 2, 0, 1, 2)$ (остальные случаи обосновываются аналогично). Тогда $(i, j) = (4k, 4l)$ или $(i, j) = (4k + 2, 4l + 2)$.

Левая вершина с цветом 0 имеет соседей в гранях, стоящих левее текущей. Левее и выше стоит грань с координатами $(i - 1, j - 1)$, следовательно, она окрашена в последовательности $(1, 0, 3, 1, \mathbf{0}, 3)$, причем рассматриваемая вершина с цветом 0 находится на 5 позиции в этом векторе (отмечено полужирным). Раскраска корректна. Правее и ниже стоит грань с координатами $(i - 1, j + 1)$, следовательно, она окрашена в последовательности $(2, 3, \mathbf{0}, 2, 3, 0)$, причем рассматриваемая вершина с цветом 0 находится на 3 позиции в этом векторе. Раскраска также корректна.

Аналогичным образом проверяем корректность раскраски для оставшихся 5 вершин текущей грани.

Двухшаговая раскраска графа $H_S(V_S, E_S)$ корректна. Для перехода к двухшаговой раскраске графа $H(V, E)$ нам фактически надо удалить все вершины из $V_S \setminus V$ вместе с инцидентными ребрами, что не повлияет на корректность раскраски.

Рассмотрим теперь раскраску графа восьмиугольной решетки $O(V, E)$ из алгоритма 3. Заметим, что она определяется на основе раскраски бесконечного графа восьмиугольной решетки $O^\infty(V^\infty, E^\infty)$. Если корректна раскраска бесконечного графа, то корректна и раскраска его конечного подграфа.

Возьмем произвольную квадратную грань графа $O^\infty(V^\infty, E^\infty)$. Верхняя вершина окрашена в цвет 0. Ее соседи:

- вершины той же грани, их цвета – 1, 2, 3;
- три нижние вершины квадратной грани, стоящей над текущей, их цвета – 1, 2, 3;
- правая вершина квадратной грани, стоящей слева от текущей, ее цвет – 1;
- левая вершина квадратной грани, стоящей справа от текущей, ее цвет – 3.

Условие двухшаговой раскраски соблюдается. Аналогичным образом оно проверяется для остальных вершин текущей грани.

Теорема 1 доказана.

5. Двухшаговая раскраска графов квадратной решетки

Определение 8 (см. [10]). Прямоугольный граф $R(n, m)$ – это подграф бесконечного графа квадратной решетки, индуцированный множеством вершин $V(n, m) = \{v \mid 1 \leq v_x \leq n, 1 \leq v_y \leq m\}$, где v_x и v_y соответствуют координатам вершины на плоскости. Граница такого графа представляет собой прямоугольник. По определению прямоугольный граф не может иметь дыр.

В статье [3] доказано, что задача о двухшаговой раскраске графа квадратной решетки в 5 цветов может быть решена за полиномиальное время. Для этого надо найти минимальный прямоугольный граф решетки $R(n, m)$, включающий в себя данный граф квадратной решетки $Q(V, E)$, и определить двухшаговую раскраску в 5 цветов графа $R(n, m)$, после чего наложить эту раскраску на граф $Q(V, E)$. Алгоритм двухшаговой раскраски (см. [3, 4]) прямоугольного графа работает за линейное время относительно количества вершин $n \cdot m$ этого графа. Более того, если $n = 2$ или $m = 2$, то прямоугольный граф решетки можно раскрасить в 4 цвета.

Произвольные графы квадратной решетки можно разделить на три класса в зависимости от максимальной степени вершины $d = \max_{v \in V}(\deg v)$:

- $d \leq 2$ – вырожденные графы квадратной решетки; правила двухшаговой раскраски рассмотрены в разделе 3;
- $d = 4$ – графы квадратной решетки, для двухшаговой раскраски которых требуется 5 цветов; раскраска может быть получена за полиномиальное время с помощью описанного выше подхода;
- $d = 3$ – графы квадратной решетки, для раскраски которых требуется 4 или 5 цветов.

Рассмотрим последний класс графов. Отметим, что в этом классе не может быть графов, для раскраски которых достаточно трех цветов (свойство 1 задачи о двухшаговой раскраске графа, см. [3]). При этом описанный выше подход позволяет найти раскраску не более чем в 5 цветов, однако эта раскраска не всегда будет оптимальной.

На рис. 4 показан граф квадратной решетки с $d = 3$, который может быть раскрашен в 4 цвета (номера цветов подписаны внутри вершин). Тем не менее, алгоритм из [3] построит двухшаговую раскраску этого графа в 5 цветов.

Однако существуют графы квадратной решетки с $d = 3$, для которых не существует двухшаговой раскраски в 4 цвета. Пример такого графа приведен на рис. 5. Нетрудно убедиться, что все четыре соседа верхней левой вершины будут окрашены попарно неравными цветами, как бы мы ни пытались окрасить остальные вершины, а значит, верхняя левая вершина должна получить цвет, не совпадающий с этими четырьмя.

Таким образом, заранее неизвестно, сколько цветов, 4 или 5, потребуется для раскраски графа квадратной решетки с $d = 3$. Более того, несмотря на кажущуюся простоту структуры такого графа, по-видимому, для него не существует быстрого алгоритма оптимальной двухшаговой раскраски. Мы рассмотрим экспоненциальный алгоритм оптимальной двухшаговой раскраски, основанный на декомпозиции графа и поиске с возвратом.

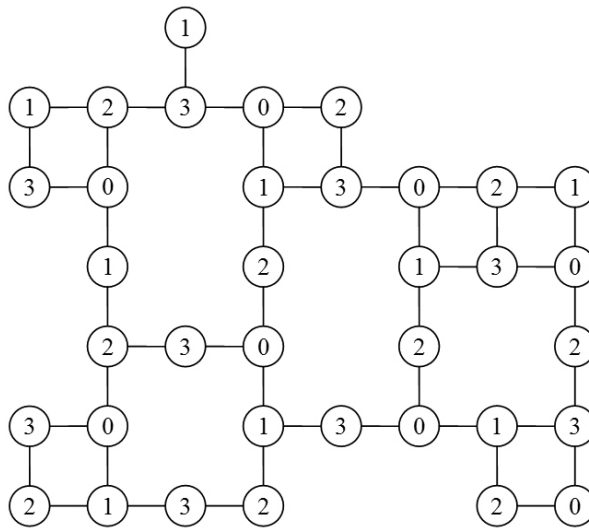


Fig. 4. Two-step colouring in 4 colours of the square grid graph

Рис. 4. Двухшаговая раскраска графа квадратной решетки в 4 цвета

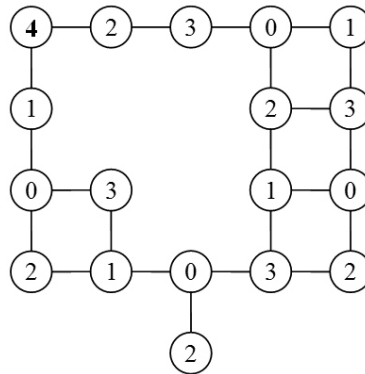


Fig. 5. Two-step colouring in 5 colours of the square grid graph

Рис. 5. Двухшаговая раскраска графа квадратной решетки в 5 цветов

Сначала проанализируем структуру графа. Можно выделить следующие составные части графа:

- прямоугольные подграфы размерности $k \times 2$ либо $2 \times l$ (одна из размерностей обязательно равна 2, иначе в подграфе будет вершина степени 4); такие подграфы требуют 4 цветов для двухшаговой раскраски, а для задания раскраски достаточно определить раскраску одной квадратной грани;
- вершины степени 3, не принадлежащие прямоугольным подграфам; имеет смысл рассматривать подграфы, образованные одной вершиной степени 3 и тремя смежными с ней вершинами; 4 вершины такого подграфа требуют 4 цветов для двухшаговой раскраски;
- вершины степени 1, не принадлежащие подграфам из предыдущего пункта;
- простые пути, соединяющие подграфы из предыдущих пунктов.

Отметим, что пара подграфов из первых двух пунктов может иметь одну или две общие вершины (за исключением случая двух прямоугольных подграфов). Вершина, являющаяся соседом прямоугольного подграфа, обязательно смежна с его угловой вершиной (не более одной смежной вершины для каждого угла), в противном случае появится вершина степени 4. Для каждого подграфа из первых двух пунктов существует 24 варианта раскраски в 4 цвета, однако какие-то раскраски пары подграфов могут противоречить друг другу.

Пусть имеется произвольный граф квадратной решетки $Q(V, E)$ с максимальной степенью вершины $d = 3$. Получим алгоритм поиска с возвратом для нахождения оптимальной раскраски такого графа. В алгоритме мы будем использовать следующие обозначения:

- A_i ($i \in \overline{1, s}$) – подграф графа $Q(V, E)$, требующий 4 цветов для двухшаговой раскраски (прямоугольный подграф либо подграф, индуцированный вершиной степени 3);
- C_i – множество потенциальных раскрасок подграфа A_i ; $0 \leq |C_i| \leq 24$; каждая раскраска – это вектор из 4 попарно различных компонент, определяющих раскраску всех вершин A_i либо одной квадратной грани A_i (в случае прямоугольного подграфа);
- v_0 – вершина, от которой на каком-либо шаге строится путь, соединяющий два подграфа A_j и A_i либо подграф A_j с висячей вершиной;
- P_0 – путь от вершины v_0 до подграфа A_i либо до висячей вершины.

Для организации возвратов используем стек, в который будет записываться номер пройденного шага и список произведенных на этом шаге окрашиваний, а также номер i рассматриваемого подграфа A_i и вершина v_0 , от которой ведется поиск пути (для соответствующих шагов).

Алгоритм 4 (оптимальная двухшаговая раскраска графа $Q(V, E)$ с $d = 3$).

1. Применим полиномиальный алгоритм из [3]: найдем оптимальную двухшаговую раскраску прямоугольного графа решетки $R(n, m)$, включающего в себя $Q(V, E)$, и наложим эту раскраску на $Q(V, E)$. Если $Q(V, E)$ раскрашен в 4 цвета, раскраска оптимальна, выход. Иначе запоминаем текущую раскраску в 5 цветов и переходим на шаг 2 для поиска раскраски в 4 цвета из множества $\{0, 1, 2, 3\}$.

2. Найдем все прямоугольные подграфы графа $Q(V, E)$, обозначим их через $A_1, \dots, A_r$, где номер 1 получает подграф с наибольшим количеством вершин, а остальные номера распределяются произвольным образом.

3. Найдем все вершины степени 3 в графе $Q(V, E)$, не принадлежащие ни одному из прямоугольных подграфов, найденных на предыдущем шаге. Для каждой такой вершины определим подграф, содержащий ее и три смежные ей вершины. Обозначим эти подграфы через $A_{r+1}, \dots, A_s$ (если на шаге 2 не найдено ни одного подграфа, нумерация идет с 1).

4. Рассмотрим подграф A_1 (такой подграф всегда есть, иначе граф вырожден). Раскрасим его в 4 цвета (если A_1 – прямоугольный подграф, используем для этого полиномиальный алгоритм из [3]) и зафиксируем раскраску. Если A_1 – прямоугольный подграф, переходим на шаг 5, иначе на шаг 6.

5. Вершинам, смежным с углами прямоугольного подграфа A_i , присваиваем цвет из $\{0, 1, 2, 3\}$, пользуясь правилами двухшаговой раскраски. Это можно сделать единственным образом. Если возникает противоречие с цветом соседа не из A_i , производим возврат (шаг 12), иначе переходим на шаг 6.

6. Берем произвольную окрашенную вершину v_0 , у которой есть неокрашенный сосед первого порядка. Если такой вершины v_0 нет, получена оптимальная раскраска в 4 цвета, выход. Если $v_0 \in A_i$ для какого-то $i \in \overline{1, s}$, полагаем $P_0 = \emptyset$ и переходим на шаг 8. Иначе ищем простой путь P_0 от v_0 через неокрашенного соседа v_0 до ближайшего подграфа A_i либо до висячей вершины. В первом случае переходим на шаг 8, во втором – на шаг 7.

7. Окрашиваем путь P_0 в 3 цвета, пользуясь правилами из раздела 3, и переходим на шаг 6.

8. Определяем множество возможных раскрасок C_i подграфа A_i . Если A_i – прямоугольный подграф, возможны варианты:

- какие-то вершины A_i уже окрашены в 3 или 4 цвета; тогда пытаемся достроить раскраску подграфа, пользуясь полиномиальным алгоритмом из [3]; если нигде не возникло противоречий, фиксируем раскраску и переходим на шаг 11; в противном случае отменяем изменения и производим возврат (шаг 12);
- какие-то вершины уже окрашены в 1 или 2 цвета; тогда пытаемся достроить частичную раскраску подграфа этими цветами, пользуясь полиномиальным алгоритмом из [3]; если

нигде не возникло противоречий, фиксируем частичную раскраску, а в C_i помещаем 6 или 2 варианта раскраски «чистых» вершин в оставшиеся 3 или 2 цвета;

- окрашенных вершин нет; множество C_i будет содержать все 24 варианта раскраски.

Если A_i – подграф, порожденный вершиной степени 3, возможны варианты:

- все вершины уже окрашены; фиксируем раскраску и переходим на шаг 11;
- 3 вершины уже окрашены; присваиваем четвертой вершине оставшийся цвет; если при этом не возникло противоречия, фиксируем раскраску и переходим на шаг 11; в противном случае отменяем изменения и производим возврат (шаг 12);
- 1 или 2 вершины уже окрашены; в C_i помещаем 6 или 2 варианта раскраски «чистых» вершин в оставшиеся 3 или 2 цвета;
- окрашенных вершин нет; множество C_i будет содержать все 24 варианта раскраски.

Переходим на шаг 9.

9. Если $C_i \neq \emptyset$, извлекаем очередной вариант раскраски из множества C_i и переходим на шаг 10. Иначе делаем возврат (шаг 12).

10. Применяем выбранную раскраску. Распространяем ее на все квадратные грани, пользуясь полиномиальным алгоритмом из [3], если A_i – прямоугольный подграф. Если нигде не возникло противоречий, фиксируем раскраску и переходим на шаг 11; в противном случае отменяем изменения и производим возврат (шаг 12).

11. Рассмотрим путь P_0 . Если A_i – прямоугольный подграф, окрашиваем вершину пути P_0 , смежную с A_i (если $P_0 \neq \emptyset$). Если в пути P_0 есть неокрашенные вершины, расширяем этот путь за счет смежных вершин так, чтобы первые две и последние две вершины пути были окрашены. После этого применяем алгоритм 1 для раскраски пути P_0 . Если удастся окрасить его в 3 или 4 цвета, фиксируем эту раскраску и переходим на шаг 5, если A_i – прямоугольный подграф, и на шаг 6 в противном случае. Если путь P_0 можно раскрасить только в 5 цветов, отменяем изменения и выполняем возврат (шаг 12).

12 (возврат). Последовательно извлекаем из стека информацию. Если это сведения об окрашивании вершины, очищаем цвет вершины. Если это номер i подграфа A_i , то делаем его текущим. Если это указание на вершину, от которой ведется поиск пути, то считаем ее вершиной v_0 . Продолжаем извлечение информации, пока не будет извлечен номер шага 9 или 4. Если извлекли 9, переходим на шаг 9. Если извлекли 4, двухшаговая раскраска графа $Q(V, E)$ в 4 цвета невозможна, оптимальной является раскраска в 5 цветов, полученная на шаге 1, выход.

Для иллюстрации принципа работы алгоритма возьмем граф с рис. 4 и выделим в нем подграфы A_i согласно алгоритму 4 (шаги 2-3). Результат показан на рис. 6, слева. Подграфы обозначены пунктирными прямоугольниками, вершины подграфа A_1 выделены серым.

Получено 10 подграфов A_i и формально мы имеем 24^{10} потенциальных вариантов для раскраски. Но на самом деле, перебор будет существенно сокращен за счет того, что цвета вершин из разных подграфов должны быть согласованы между собой. Действительно, раскрасим подграф A_1 и смежные с ним вершины, применив шаги 4-5 алгоритма. Нетрудно убедиться, что раскраска шести вершин подграфа A_i однозначно определит цвет еще 7 вершин, 6 из которых принадлежат другим A_i (рис. 6, справа). Так, например, для трех верхних подграфов глубина перебора будет сокращена с 13 824 до 8 вариантов раскраски. Продолжая выполнение алгоритма, мы за приемлемое время дойдем до двухшаговой раскраски, показанной на рис. 4.

6. Двухшаговая раскраска графов треугольной решетки

Графы треугольной решетки наиболее сложны в плане двухшаговой раскраски среди всех графов решетки. Это связано с тем, что в общем случае максимальная степень вершины в таком графе равна 6. Это значит, что нам потребуется не менее 7 цветов для двухшаговой раскраски графа треугольной решетки, если в нем есть хотя бы одна вершина степени 6. Пусть имеется произвольный

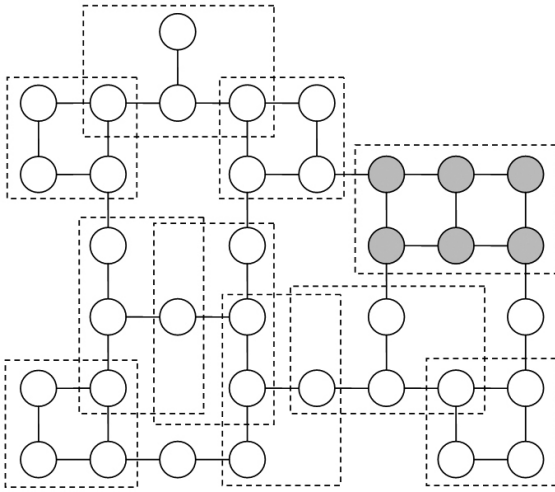


Fig. 6. Selection of the sets A_i from the square grid graph

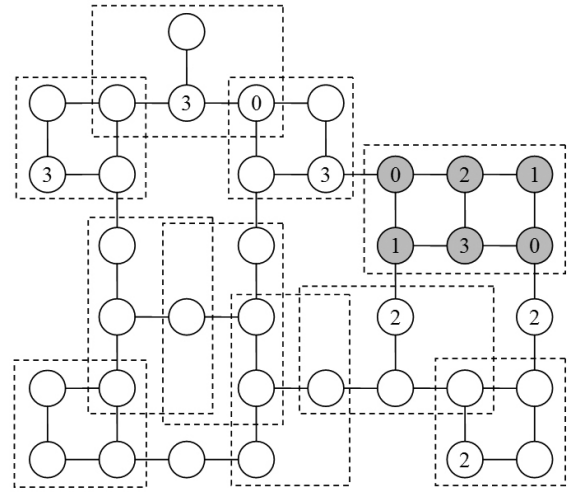


Рис. 6. Выделение множеств A_i в графе квадратной решетки

невырожденный граф треугольной решетки $T(V, E)$. Рассмотрим алгоритм, реализующий такую раскраску за линейное относительно количества вершин время.

Алгоритм 5 (двухшаговая раскраска графа треугольной решетки в 7 цветов).

1. Возьмем произвольную вершину из V и присвоим ей цвет 0. Все вершины из V не просмотрены.

2. Возьмем произвольную окрашенную и непросмотренную вершину $v_0 \in V$. Пусть ее цвет равен c_0 . Окрасим неокрашенные смежные с v_0 вершины по следующим правилам:

- вершина слева от v_0 получает цвет $(c_0 - 1) \bmod 7 = (c_0 + 6) \bmod 7$;
- вершина справа от v_0 получает цвет $(c_0 + 1) \bmod 7$;
- вершина слева-вверху от v_0 получает цвет $(c_0 - 3) \bmod 7 = (c_0 + 4) \bmod 7$;
- вершина справа-внизу от v_0 получает цвет $(c_0 + 3) \bmod 7$;
- вершина справа-вверху от v_0 получает цвет $(c_0 - 2) \bmod 7 = (c_0 + 5) \bmod 7$;
- вершина слева-внизу от v_0 получает цвет $(c_0 + 2) \bmod 7$.

3. Вершина v_0 становится просмотренной. Повторяем шаг 2, пока есть непросмотренные вершины.

Рис. 7 (слева) иллюстрирует шаг 2 алгоритма 5. Здесь для вершины, отмеченной нулем, показаны все возможные соседи первого и второго порядка, для каждого из которых указана величина, на которую его цвет будет отличаться от цвета центральной вершины (по модулю 7).

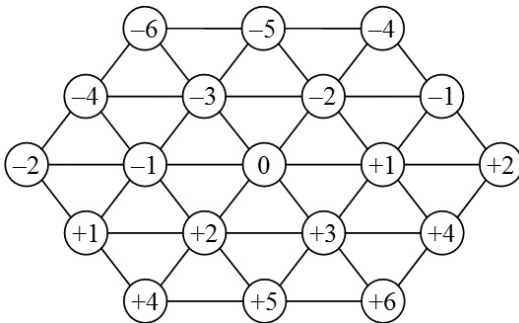


Fig. 7. Rules for counting neighbours' colours and two-step colouring in 7 colours of the triangular grid graph

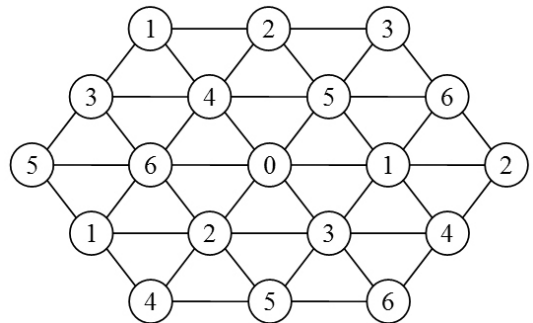


Рис. 7. Правила вычисления цвета соседей и двухшаговая раскраска графа треугольной решетки в 7 цветов

Справа на рис. 7 показан пример двухшаговой раскраски графа треугольной решетки, полученной с помощью алгоритма 5.

Теорема 2. *Двухшаговая раскраска любого невырожденного графа треугольной решетки в 7 цветов всегда может быть найдена с помощью полиномиального алгоритма 5.*

Доказательство. Для доказательства теоремы достаточно показать, что двухшаговая раскраска, полученная с помощью алгоритма 5, корректна. Рис. 7 наглядно демонстрирует корректность раскраски всех соседей любой выбранной вершины. Остается заметить, что порядок обхода вершин не имеет значения: каким бы путем мы не пришли из вершины, окрашенной на шаге 1, в любую выбранную, цвет для выбранной вершины будет вычислен одинаково (противоположно направленные переходы по ребрам дают взаимно обратные изменения цвета, полный обход любого треугольника и, как следствие, любого цикла приводит к исходному цвету).

Теорема 2 доказана.

Отметим, что алгоритм 5 реализует оптимальную раскраску для всех графов, в которых есть хотя бы одна вершина степени 6. Во всех остальных случаях решение может отличаться от оптимального. Более того, для нахождения оптимальной двухшаговой раскраски графа треугольной решетки с ограничением на степень вершины, вероятней всего, потребуется экспоненциальный алгоритм поиска с возвратом, подобный алгоритму 4.

References

- [1] S. V. Korsakov, A. V. Smirnov, and V. A. Sokolov, “Principles of Organizing the Interoperability of Equipollent Nodes in a Wireless Mesh-Network with Time Division Multiple Access”, *Automatic Control and Computer Sciences*, vol. 50, no. 6, pp. 415–422, 2016. DOI: [10.3103/S0146411616060031](https://doi.org/10.3103/S0146411616060031).
- [2] F. Harary, *Graph theory*. Addison-Wesley Pub. Co., 1969.
- [3] N. S. Medvedeva and A. V. Smirnov, “NP-Completeness and One Polynomial Subclass of the Two-Step Graph Colouring Problem”, *Automatic Control and Computer Sciences*, vol. 54, no. 7, pp. 685–696, 2020. DOI: [10.3103/S0146411620070159](https://doi.org/10.3103/S0146411620070159).
- [4] N. S. Medvedeva and A. V. Smirnov, “Dvukhshagovaya raskraska pryamougol’nogo grafa reshetki”, in *Zametki po informatike i matematike*, vol. 11, Yaroslavl: YSU, 2019, pp. 131–138.
- [5] A. N. Kolmogorov, “Parkety iz pravil’nykh mnogougol’nikov”, *Kvant*, no. 3, pp. 24–27, 1970.
- [6] C. Umans and W. Lenhart, “Hamiltonian Cycles in Solid Grid Graphs”, in *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS)*, 1997, pp. 496–505. DOI: [10.1109/SFCS.1997.646138](https://doi.org/10.1109/SFCS.1997.646138).
- [7] A. Itai, C. H. Papadimitriou, and J. L. Szwarcfiter, “Hamilton Paths in Grid Graphs”, *SIAM Journal on Computing*, vol. 11, no. 4, pp. 676–686, 1982. DOI: [10.1137/0211056](https://doi.org/10.1137/0211056).
- [8] V. S. Gordon, Y. L. Orlovich, and F. Werner, “Hamiltonian properties of triangular grid graphs”, *Discrete Mathematics*, vol. 308, no. 24, pp. 6166–6188, 2007. DOI: [10.1016/j.disc.2007.11.040](https://doi.org/10.1016/j.disc.2007.11.040).
- [9] K. Islam, H. Meijer, Y. Núñez, D. Rappaport, and H. Xiao, “Hamilton Circuits in Hexagonal Grid Graphs”, in *Proceedings of the 19th Canadian Conference on Computational Geometry (CCCG2007)*, 2007, pp. 85–88.
- [10] F. Keshavarz-Kohjerdi and A. Bagheri, “An efficient parallel algorithm for the longest path problem in meshes”, *The Journal of Supercomputing*, vol. 65, no. 2, pp. 723–741, 2013. DOI: [10.1007/s11227-012-0852-0](https://doi.org/10.1007/s11227-012-0852-0).