

«Раскраска 3-раскрашиваемого графа»
ФПМИ МФТИ

Тармаев Александр Б05-122

Зима 2024

Аннотация

Задача о раскраске графа (graph coloring problem) заключается в нахождении по данному графу раскраски его вершин с использованием минимального количества цветов так, чтобы концы любого ребра были покрашены в разные цвета. Известно, что задача покраски графа $G = \langle V, E \rangle$ в k цветов является NP-полной. Однако если известно, что граф является 3-раскрашиваемым, то нетрудно предоставить алгоритм, способный раскрасить его в $O(\sqrt{n})$ цветов, где n - число вершин в графе.

Содержание

1	Введение	3
2	Основные понятия и утверждения	3
2.1	Определения и постановка задачи	3
2.2	NP-трудность	3
2.3	Алгоритм раскраски	3
2.3.1	Описание	4
2.3.2	Корректность	4
2.3.3	Асимптотика	4
3	Выводы	4
	Список литературы	5

1 Введение

Правильной вершинной раскраской графа G называется произвольное отображение $c: V(G) \rightarrow N$ такое, что $c(v_1) \neq c(v_2) \forall$ смежных вершин $v_1, v_2 \in V(G)$. Правильная вершинная раскраска c графа G называется k -раскраской, если $c: V(G) \rightarrow 1, k$. Если граф G имеет k -раскраску, то он называется k -раскрашиваемым. Хроматическим числом графа G называется такое наименьшее k , что граф G является k -раскрашиваемым. Оно обозначается через $\chi(G)$. Задача о вершинной раскраске для заданных графа G и числа k состоит в том, чтобы определить, выполняется ли неравенство $\chi(G) \leq k$ или нет. Задача о вершинной k -раскраске (кратко, задача k -BP) для заданного графа G заключается в том, чтобы определить, выполняется ли неравенство $\chi(G) \leq k$ или нет. Обе задачи являются классическими NP-полными задачами на графах. Если же нам известно, что G 3-BP, при этом сама раскраска нам неизвестна, то мы можем раскрасить его, применив корневую декомпозицию за полиномиальное время.

2 Основные понятия и утверждения

2.1 Определения и постановка задачи

Def. Классом $\text{NTIME}(T(n))$ называется класс языков, которые распознаются на недетерминированной машине Тьюринга за время $O(T(n))$.

Def. $\text{NP} = \bigcup_{i=1}^{\infty} \text{NTIME}(n^i)$

Def. Язык $A \leq_p$ языка B , если \exists вычислимая f , такая что $x \in A \Leftrightarrow f(x) \in B$

Def. Язык B является NP-трудным, если для любого $A \in \text{NP}$ выполнено $A \leq_p B$. Язык B является NP-полным, если он NP-трудный и лежит в NP.

2.2 NP-трудность

В общем случае задача раскраски графа в k цветов является NP-трудной задачей. $\in \text{NP}$, так как сертификатом может являться искомая раскраска вершин по цветам, а $\in \text{NPH}$, так как сводима к 3КНФ-SAT (См. список литературы [1])

2.3 Алгоритм раскраски

Лемма. Граф с максимальной степенью вершины d можно раскрасить в $d+1$ цвет.

Докажем индукцией по количеству вершин:

База: $n=1, k=0$, граф является 1-раскрашиваемым.

Переход: Пусть доказано для n , докажем для графа с $n+1$ вершиной и максимальной степенью k . Уберём из графа 1 вершины с инцидентными рёбрами, оставшиеся компоненты графа можно раскрасить в k цветов, тогда добавив рёбра обратно сможем раскрасить вершину в отличный от соседей цвет.

Давайте предъявим алгоритм раскраски 3-раскрашиваемого графа в $O(\sqrt{n})$ цветов

2.3.1 Описание

Algorithm 1 Алгоритм раскраски

1. Перебираем все вершины от 0 до $n-1$, если её степень хотя бы \sqrt{n} , то покрасим всех её соседей в 2 новых цвета, удалим из графа
 2. Когда таких вершин не осталось, то по лемме 1 можем раскрасить вершины графа в \sqrt{n} цвет, что мы и сделаем, запустив рекурсивный алгоритм.
-

2.3.2 Корректность

Всех соседей произвольной вершины в 2 цвета, так как граф является 3-раскрашиваемым, раскрасить оставшиеся вершины можем по лемме. Так как вершин степени $\geq \sqrt{n}$ не больше \sqrt{n} , то будет использовано $\leq 2\sqrt{n}$ цветов. Затем так как останутся вершины степени $< \sqrt{n}$, то из можно будет раскрасить в $\leq \sqrt{n}$ цветов.

2.3.3 Асимптотика

Утверждение. Алгоритм работает за $O(n^2)$.

Доказательство: На первом шаге если степень вершины хотя бы \sqrt{n} , то за линию можем удалить рёбра и покрасить соседей. Итого так как таких вершин не более \sqrt{n} , то сможем сделать за $O(n^{1.5})$. Затем раскрашиваем оставшиеся вершины, запускаем алгоритм dfs в какое-то время, то есть сложность $O(n^2)$.

3 Выводы

Изучили метод неоптимальной, но быстрой раскраски графа в не очень большое число цветов при заданных условиях. Такими предположениями и ослаблениями NP-трудная задача превратилась в задачу сложности $O(n^2)$

Список литературы

- [1] Алгоритмы для NP-трудных задач https://www.lektorium.tv/sites/lektorium.tv/files/additional_files/20091115_algorithmsfornphardproblems_kulikov_lecture09.pdf
- [2] NP-полнота задачи о раскраске графа, ИТМО https://neerc.ifmo.ru/wiki/index.php?title=NP-%D0%BF%D0%BE%D0%BB%D0%BD%D0%BE%D1%82%D0%B0_%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8_%D0%BE_%D1%80%D0%B0%D1%81%D0%BA%D1%80%D0%B0%D1%81%D0%BA%D0%B5_%D0%B3%D1%80%D0%B0%D1%84%D0%B0