Раскраска вершин графа в минимальное количество цветов без одноцветных ребер

Автор проекта: Аникин Матвей

Аннотация

Рассматривается задача раскраски вершин неориентированного графа минимально возможным числом цветов. Описан и проанализирован простой алгоритм, реализованный автором. На каждом шаге выбирается нераскрашенная вершина с максимальной *насыщенностью* — количеством различных цветов среди уже раскрашенных соседей (при равенстве берётся вершина большей степени). Выбранная вершина получает наименьший не конфликтующий цвет, после чего информация о насыщенности соседей обновляется. Доказана корректность алгоритма (всегда получается правильная раскраска), установлена верхняя оценка на число цветов $\Delta(G) + 1$, где $\Delta(G)$ — максимальная степень графа, и выполнен анализ трудоёмкости и инвариантов реализации.

1 Введение

Пусть задан конечный неориентированный граф $G = (V, E)$. Требуется окрасить вершины так, чтобы концы любого ребра имели разные цвета. Минимальное количество цветов называется хроматическим числом графа.

В данной работе рассматривается конкретный жадный алгоритм. Идея следующая: на каждом шаге сначала выбирается наиболее "сложная" для покраски вершина, то есть та, у которой среди уже окрашенных соседей встречается наибольшее число различных цветов (это значение назовём насыщенностью). Интуитивно такую вершину лучше раскрашивать раньше, пока палитра свободнее. После выбора вершине присваивается минимальный цвет, не встречающийся у её окрашенных соседей. Далее обновляется информация о цветах, встречающихся у соседей, и процесс повторяется до тех пор, пока не будут окрашены все вершины.

2 Техническая часть

2.1 Определения

Определение 1. Правильной раскраской графа $G = (V, E)$ в k цветов называется отображение $c : V \to \{0, 1, \dots, k-1\}$ такое, что для всех $\{u, v\} \in E$ выполнено $c(u) \neq c(v)$.

Определение 2. Для частично построенной раскраски c (часть вершин может быть не окрашена) *насыщенностью* неокрашенной вершины v называется мощность множества цветов, встречающихся у уже окрашенных соседей v:

$$\mathrm{sat}(v) = \Big|\{\, c(u) \;:\; u \in N(v),\; c(u) \neq \bot \,\}\Big|.$$

Здесь символ $\perp$ означает "не окрашено".

2.2 Описание алгоритма

Пусть граф задан списками смежности. Алгоритм поддерживает:

- массив $c[v] \in \{-1, 0, 1, \dots\}$ — цвет вершины (-1 означает "не окрашена");
- для каждой вершины множество $S[v]$ — множество цветов, встречающихся среди уже окрашенных соседей v; его мощность равна насыщенности.

Итерация алгоритма:

1. Выбор вершины. Среди всех v с $c[v] = -1$ выбирается вершина с максимальной насыщенностью $|S[v]|$. При равенстве выбирается вершина с большей степенью $|N(v)|$.
2. Присвоение цвета. Вычисляется наименьшее неотрицательное целое t, которого нет в $S[v]$. Этим цветом красится v, то есть $c[v] \leftarrow t$.
3. Обновление. Для каждого соседа $u \in N(v)$, ещё не окрашенного ($c[u] = -1$), в множество $S[u]$ добавляется цвет t.

Процесс продолжается, пока не окрашены все вершины.

2.3 Инвариант

Лемма 1. *В любой момент выполнения алгоритма для каждой неокрашенной вершины v множество $S[v]$ в точности совпадает с множеством цветов у уже окрашенных соседей v.*

Доказательство. Инвариант верен изначально, так как все вершины не окрашены, и все $S[v] = \varnothing$. При окраске вершины v единственное изменение касается соседей $u \in N(v)$: к их множеству цветов корректно добавляется назначенный цвет $c[v]$. На остальных вершинах множество S не меняется. Следовательно, инвариант сохраняется по индукции по числу шагов. □

3 Основная часть

3.1 Корректность

Теорема 1. *Алгоритм завершает работу за конечное число шагов и возвращает правильную раскраску графа.*

Доказательство. На каждом шаге окрашивается ровно одна ранее неокрашенная вершина, а их конечное число $|V| = n$, значит алгоритм завершится не более чем за n шагов.

Покажем правильность. Рассмотрим момент окраски очередной вершины v. По определению цвета t он выбирается минимальным неотрицательным, отсутствующим в $S[v]$. По лемме $S[v]$ содержит все цвета окрашенных соседей v, следовательно t не совпадает ни с одним из них, а неокрашенные соседи цвета пока не имеют. Значит, ни на одном ребре, инцидентном v, конфликт не возникает. Ранее окрашенные вершины сохраняют свои цвета, поэтому вся построенная часть раскраски остаётся правильной. После n шагов правильно окрашен весь граф. □

3.2 Оценка числа цветов

Теорема 2. *Пусть* $\Delta = \max_{v \in V} \deg(v)$*. Алгоритм использует не более* $\Delta+1$ *цветов.*

Доказательство. При окраске вершины v множество $S[v]$ содержит цвета не более чем $\deg(v) \le \Delta$ соседей. Следовательно среди цветов $\{0, 1, \dots, \Delta\}$ всегда найдётся свободный, в частности минимальный отсутствующий. Поэтому никогда не требуется цвет с номером $\Delta + 1$ и выше. $\square$

3.3 Трудоёмкость

Обозначим $n = |V|$, $m = |E|$.

- Выбор вершины на каждой итерации выполняется линейным просмотром всех n вершин, поэтому суммарно $O(n^2)$.
- Обновление множеств $S[u]$ за всю работу алгоритма делает по одной вставке на каждое ориентированное ребро, то есть $O(m)$ операций вставки в множество (в реализации — амортизированно $O(1)$).
- Поиск минимального свободного цвета для вершины v занимает $O(|S[v]|) \le O(\deg(v))$. Сумма по всем вершинам — $O(\sum_v \deg(v)) = O(m)$.

Итого асимптотика $O(n^2 + m)$ по времени и $O(n + m)$ по памяти (списки смежности плюс множества цветов соседей).