В.А. Евстигнеев, Ы. Турсунбай кызы

ДИНАМИЧЕСКИЙ РАСПРЕДЕЛЕННЫЙ ПН-АЛГОРИТМ ДЛЯ РАСКРАСКИ w-СОВЕРШЕННЫХ ГРАФОВ

Данная работа посвящена раскраске w-совершенных графов в рамках распределенной модели вычислений, которая использует широко известную стратегию ПН-алгоритма. Класс w-совершенных графов довольно широкий и содержит в себе такие практически интересные классы графов, как, например, класс хордальных графов [6], который является одним из наиболее изученных и широко применяемых классов графов.

ВВЕДЕНИЕ

Рассмотрим задачу раскраски вершин графа в рамках распределенной модели вычислений. Распределенные вычисления на графах представляют собой такую организацию вычислений, при которой отсутствует всякая возможность использовать глобальные операции и механизмы, а также возможность получать информацию иначе, нежели используя информацию из локальной памяти соседей. Это означает, что в основе любого распределенного вычисления на графах лежит организация взаимодействия с соседними вершинами и пересылка локальной информации.

Область применения распределенных вычислений на графах — организация целенаправленной деятельности коллектива исполнителей (автономных устройств, ЭВМ в составе сети, распределенной вычислительной системы и т.п.) путем обмена сообщениями между «близкими» в некотором смысле членами коллектива и при отсутствии каких-либо глобальных механизмов.

Данная распределенная модель раскраски может быть использована в распределенных беспроводных сетях для устранения столкновений пакетов путем назначения ортогональных кодов радиостанциям [1].

Заметим, что не всегда легко добиться и скорости, и эффективности алгоритма. В работе [2] представлен распределенный алгоритм для раскраски графа в (Δ +1) цветов, где Δ — наибольшая степень вершины в графе. Время работы алгоритма O(log n). Будем называть этот алгоритм тривиальным. Данный алгоритм достаточно простой и быстрый, но не оптимальный. Действительно, количество цветов, используемых алгоритмом, близок к Δ, даже если граф двудольный. Неудивительно, что тривиальный алгоритм не имеет механизма экономии цветов. Дальнейшее усовершенствование тривиального алгоритма предложен в [3]. В данной работе приведен новый алгоритм для раскраски в O (Δ / log Δ) цвета, но этот алгоритм работает только на графах без триангуляторов.

Одним из способов улучшения выполнения распределенного алгоритма является представление стратегии раскраски в алгоритм, который, как известно, является эффективным в нераспределенных алгоритмах.

В работе [4] представлен распределенный алгоритм для раскраски вершин графа, который походит на стратегию Наибольшие Первые. Раскраска, полученная с помощью данного алгоритма оптимальна или близка к оптимальному для некоторых классов графов, таких как, полные k-сторонние, гусеницы, короны и двусторонние колеса.

В настоящей работе исследуется задача раскраски w-совершенных графов с помощью ПН-алгоритма, представлен динамический распределенный алгоритм для решения данной задачи.

1. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Раскраской вершин графа G называется такое приписывание цветов его вершинам, что никакие две смежные вершины не получают одинаковые цвета. *Хроматическое число* χ (G) графа G определяется как наименьшее количество цветов, необходимых для раскраски G, а раскраска G χ (G) цветами называется *оптимальной раскраской* графа G. Задача раскраски графа заключается в нахождении оптимальной раскраски. Характерной особенностью этих задач является существование объектов, которые по каким-либо причинам не могут быть объединены в одну группу.

Поскольку задача определения хроматического числа принадлежит классу полиномиально полных задач, исследования в этой области ведутся в разных направлениях. Принципиальные трудности, которые возникают при раскраске графа и нахождении его хроматического числа, вынуждают, во-первых, найти и исследовать практически интересные классы графов, для которых задача раскраски полиномиально разрешима, и, во-вторых, вычислить или оценить хроматическое число графа с помощью других, более легко вычисляемых характеристик графа.

Одной из важных характеристик, связанных с хроматическим числом, является число Визинга—Вилфа $w(G) = \max_{G' \subset G} \delta(G') + 1$, где

$\delta(G') = \min\limits_{x \in V(G')} d_{G'}(x)$ — минимальная степень графа G', а $d_{G'}(x)$ — степень вершины x в G'. $w(G)$ в качестве верхней оценки хроматического числа впервые рассмотрена в работе Секереша и Вилфа [5].

Важность этой характеристики заключается в том, что, во-первых, $w(G)$ является довольно нетривиальной верхней оценкой для $\chi(G)$ [5], т.е. класс графов, для которых $w(G) = \chi(G)$, довольно большой и содержит в себе много практически интересных классов, и, во-вторых, она легко вычисляемая.

Определение. Граф, обладающий тем свойством, что хроматическое число и вырожденность (число Визинга—Вилфа) равны не только у самого графа, но и у каждого его порожденного подграфа называется w-*совершенным графом*.

Важным подклассом w-совершенных графов являются хордальные графы [6].

Определение. Граф называется *хордальным (триангулированным)*, если каждый его цикл длины > 3 содержит хорду, т.е. ребро, соединяющее несмежные вершины простого цикла.

Более подробные определения и свойства w-совершенных графов приведены в работе [7, 8].

Для вычисления $w(G)$ вводится понятие упорядочения по наименьшему последнему (*ПН-упорядочение*) [9, 10, 11] графа G. Оно строится следующим образом:

а) для $n = n(G)$ в качестве v_i выбирается вершина минимальной степени в графе G;

б) для $i = 2, 3, ..., n$ в качестве v_i выбирается вершина минимальной степени в подграфе $G \setminus \{v_1, ..., v_{i-1}\}$.

Для упорядоченного множества вершин $v_1, ..., v_n$ графа G последовательной раскраской, отвечающей этому порядку, называется раскраска, определяемая следующим образом:

а) вершине v_n приписан цвет 1;

б) для $i = n-1, ..., 1$ вершина v_i получает цвет с наименьшим номером, не встречающийся на смежных с вершиной v_i вершинах.

Процедура определения ПН-упорядочения вершин и нахождения по нему раскраски называется *ПН-алгоритмом* [9]. По своему строению ПН-алгоритм приводит к раскраске не более чем $w(G)$ цветами.

2. ОПИСАНИЕ АЛГОРИТМА

Определение. *Параллельно-последовательным вычислительным алгоритмом* называется локальный алгоритм, каждый шаг которого состоит в обработке параллельно и независимо всех (или некоторых) вершин, смежных с вершинами, обработанными на предыдущем шаге; на первом шаге обрабатывается фиксированное множество начальных вершин [12].

Теорема. Задача раскраски w-совершенных графов ПН-алгоритмом разрешима в классе распределенных параллельно последовательных алгоритмов.

Доказательство:
Рассмотрим описание соответствующего распределенного алгоритма.
Мы предположим, что система синхронизирована в раундах.
Каждая вершина имеет следующие параметры:

– случайное значение: ***rndvalue (v);***
– номер, соответствующий ПН-упорядочению: ***SLnumber (v)*** (в начале номера всех вершин равны единице, т.е. SLnumber (v)=1);
– показатель состояния параметра SL-number: ***cond (v),*** который имеет либо значение I (intermediate) — промежуточное, либо значение F (final) — конечное (в начале все вершины имеют промежуточное состояние);
– количество соседних вершин, для которых не установлены конечные номера, т.е. cond (v) = I: ***ddeg (v)*** (в начале ddeg (v) = deg (v));
– палитра «запрещенных» цветов, цвета, которые были использованы соседними вершинами: ***usedcolor (v)*** (в начале пустая).

Пусть $v_1, v_2 \in V$. Мы говорим, что вершина v_1 имеет более высокий приоритет чем v_2, если: ddeg (v_1) < ddeg (v_2) или (ddeg (v_1) = ddeg (v_2)) и (rndvalue (v_1) < rndvalue (v_2)).

В каждом раунде все неокрашенные вершины параллельно и независимо друг от друга проделывают следующее:

1. Вершина v выбирает параметр rndvalue (v) $\in [0..1]$;

2. Посылает всем соседям следующие параметры: ddeg (v), rndvalue (v);

3. Сравнивает свои параметры с полученными от соседей и проверяет, какая вершина имеет более высокий приоритет;

4. Если вершина v имеет высокий приоритет, то она оставляет себе текущее значение параметра SLnumber и меняет значение параметра cond (v) с промежуточного на конечный. В противном случае, увеличивает на единицу значение параметра SLnumber (v);

5. Пересчитывает параметр ddeg (v).

6. Если ddeg (v) = 0, т.е. всем смежным вершинам назначены конечные ПН-номера, увеличивает на единицу значение параметра SLnumber и меняет значение параметра cond (v) с промежуточного на конечный, переход к шагу 7, иначе переход к шагу 2;

7. Посылает всем соседям параметры: SLnumber (v) и первый предполагаемый цвет с наименьшим номером (не находящийся в списке «запрещенных»);

8. Сравнивает свои параметры с полученными от соседей, проверяет, какая вершина имеет наибольший SLnumber, если вершина имеет наибольший номер, то оставляет предполагаемый цвет и стоп.

9. В противном случае обновляет список usedcolor (v), переход к шагу 6.

ПН-алгоритм можно условно разделить на два этапа:

1. Каждой вершине, имеющей минимальную степень, устанавливается номер, соответствующий ПН — упорядочению (1–6 шаг);
2. Производится раскраска графа G, начиная с вершин, которые имеют большие ПН-номера (7–9 шаг).

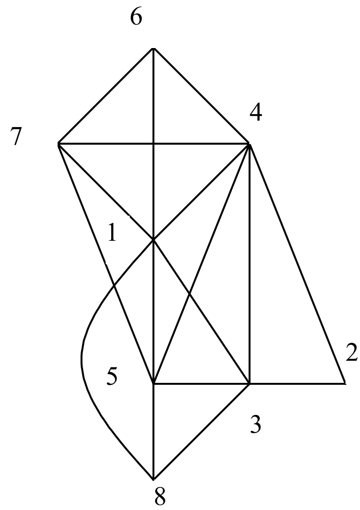

Рис. 1. w-совершенный граф

Рассмотрим пример, показанный на рис. 1.

В начале всех раундов граф находится в следующем состоянии: каждая вершина имеет параметр ddeg (v), значением которого является степень данной вершины; SL номера всех вершин равны 1; состояния данных параметров является промежуточными; ни одна вершина не окрашена, и список «запрещенных» цветов пуст.

В начале алгоритма все вершины выбирают себе значение параметра rndvalue из интервала [0..1]. В первом раунде конечное состояние параметра SLnumber примут вершины 2 (ddeg (2) = 2), 6 (ddeg (2) = 3) и 8 (ddeg (2) = 3), которые имеют наименьшие степени в своей окрестности и соответственно имеют высокий приоритет. Данные вершины не являются соседними и «не влияют» друг на друга, поэтому все они сохраняют начальное значение параметра SLnumber (v) = 1. Остальные вершины увеличат на единицу значение данного параметра и пересчитывают значение ddeg (v).

Во втором раунде SL номера получат вершины 3 (ddeg (2) = 3) и 7(ddeg (2) = 3).

В третьем раунде смежные вершины 1, 4 и 5 имеют одинаковые параметры ddeg и в данном случае приоритет той или иной вершины определяет случайное значение rndvalue. В этом раунде конечный SLnumber получит вершина 1 (rndvalue = 1.09, ddeg (4) = 2).

В четвертом раунде конечный SLnumber будет назначен вершине 4 (rndvalue = 1.38б ddeg (1) = 1) и в пятом раунде вершине 5 (rndvalue = 1.62, ddeg (5) = 0).

После окончания каждого раунда для каждой вершины проверяется условие: ddeg (v) = 0. Если ответ положительный, тогда вершина переходит ко второму этапу алгоритма, в противном случае продолжает выполнять операции первого этапа.

На втором этапе вершина, для которой ddeg (v) = 0 отсылает всем соседям свой SLnumber и первый предполагаемый цвет с наименьшим номером. Необходимо отметить, что вершина не может быть окрашена в предполагаемый цвет до тех пор, пока данное условие не будет выполнено для всех вершин окрестности 2го порядка.

В 6-м раунде вершина 5, которая имеет наибольший SLnumber среди соседних вершин, будет окрашена в предполагаемый цвет с номером 1. Данная вершина останавливает все свои действия. Остальные вершины обновляют список «запрешенных» цветов usedcolor. Неокрашенные вершины продолжают аналогичные действия. После 10го раунда распределение цветов будет выглядеть следующим образом: вершина 4 (цвет — 2, раунд — 7), вершина 1 (цвет — 3, раунд — 8), вершины 3, 7 (цвет — 4, раунд — 9), вершины 2, 6 (цвет — 1, раунд — 10), вершина 1 (цвет — 2, раунд — 10).

СПИСОК ЛИТЕРАТУРЫ

1. **Battiti R., Bertossi A.A., Bonuccelli M.A.** Assigning codes in wireless networks // Wireless Networks 5. — 1999. — P. 195–209.
2. **Johansson Ö.** Simple distributed $\Delta + 1$ — coloring of graphs // Inf. Process. Letters. — 1999. — Vol. 70. — P. 229–232.
3. **Grable D.A., Panconesi A.** Fast distributed algorithms for Brooks-Vizing colorings // J. Algorithms 37. — P. 85–120.
4. **Hansen J., Kubale M., Kuszner L. Nadolski A.** Distributed Largest-first algorithm for graph coloring // Proc. of EuroPar 2004. — Lect. Notes Comput. Sci. — 2004. — Vol. 3149. — P. 527–539.
5. **Szekeres G., Wief H.S.** An inequality for the chromatic number of a graph // J. Combin. Theory. — 1964. — Vol. 4. — P. 1–3.
6. **Волошин В.И.** Свойство триангулированных графов // Исслед. операций и программирования мат.наук. — Кишинев, 1982. — С. 24–32.
7. **Маркосян С.Е., Гаспарян Г.С.** w-совершенные графы // Ученые записки. — Ереван.гос.универ-т, 1987. — № 3. — С. 9–15.
8. **Евстигнеев В.А.** Хордальные графы и их свойства // Проблемы систем информатики и программирования. — Новосибирск, 1999. — С. 33–64.
9. **Евстигнеев В.А.** Применение теории графов в программировании. — М.: Наука, 1985. — 352 с.
10. **Кристофиди Н.** Теория графов. Алгоритмический подход. — М.: Мир, 1978. — 432 с.
11. **Matula D.W., Bleck L.L.** Smallest-last ordering and dustering and graph coloring algorithms // J. Assoc. Comput. Math. — 1983. — Vol. 30. N 3. — P. 417–427.
12. **Евстигнеев В.А.** О некоторых свойствах локальных алгоритмов на графах // Комбинаторно-алгебраические методы в прикладной математике. — Горький: ГГУ, 1983. — С. 72–105.