

3. Помимо локальности вычислений, повышению быстродействия способствует
1. Калантаров П. Л., Цейтлин Л. А. Расчет индуктивностей: Справочная книга.— Л.: Энергоатомиздат, 1986.
 2. Ерофеева О. В., Савенков В. П. Улучшение метрологических характеристик устройств графического ввода // Автометрия.— 1985.— № 2.
 3. А. с. 739575 СССР. Устройство для считывания графической информации/В. А. Акапович, Г. И. Алексеев, Б. Я. Жевелев и др.— Оцубл. 5.06.80. Бюл. № 21.
 4. Алексеев Г. И. Электромагнитные планшетные устройства ввода/Под ред. П. М. Чеголина.— Минск: Наука и техника, 1985.
 5. Ермаков С. Л., Жевелев Б. Я., Рудой В. А. Указатель типа «карандаш» в координатно-измерительных средствах повышенной точности // Автометрия.— 1985.— № 3.
 6. Чеголин П. М., Леонович Э. Н., Савенков В. П. Автоматизация преобразования сложных форм графической информации.— Минск: Наука и техника, 1973.

Поступило в редакцию 22 декабря 1987 г.

УДК 681.3.06 : 519.17

Е. М. ХЕЙФЕЦ

(Рига)

АЛГОРИТМ РАСКРАСКИ ГРАФА С САМООБУЧЕНИЕМ

Разнообразные проблемы, встречающиеся при составлении расписаний, распределении ресурсов, проектировании печатных плат, в программировании и т. д., могут быть сведены к задаче раскраски вершин конечного неориентированного графа в минимальное число цветов. Эта задача является *NP*-полной, поэтому усилия многих исследователей направлены на конструирование достаточно эффективных методов приближенного ее решения.

В основе большинства подобных методов лежит так называемый последовательный алгоритм, суть которого заключается в следующем. Пусть элементы множества вершин графа $V = \{v_1, v_2, \dots, v_n\}$ и разрешенных для присвоения цветов $C = \{c_1, c_2, \dots, c_m, \dots\}$ пронумерованы. Последовательный алгоритм включает следующие действия:

- а) вершине v_1 присваивается цвет с первым номером c_1 ;
- б) каждой последующей вершине v_i , $i = 2, 3, \dots, n$, присваивается цвет c_j с наименьшим номером, не встречающимся на смежных с v_i вершинах. Будем называть глобально оптимальным решением задачи раскраски такое присвоение $A: V \rightarrow C$, для которого $\text{card}(V/A) = \chi$, где χ — хроматическое число графа. Если вершины графа правильно пронумерованы, то с помощью последовательного алгоритма можно найти глобально оптимальное решение. Однако алгоритм правильной нумерации неизвестен.

Описанные в литературе приближенные методы решения задачи раскраски отличаются друг от друга в основном способами упорядочения вершин графа [1]. Основная идея упорядочения состоит в том, чтобы поместить в начало перечня те вершины, которым на заключительной стадии работы последовательного алгоритма могут понадобиться цвета с большими номерами. Такие вершины будем называть трудными для присвоения. В качестве степени трудности присвоения принято использовать степени $\delta(v_i)$ раскрашиваемого графа, степени вершин определенным образом выделяемых подграфов, число различных цветов на смежных с v_i вершинах в частично раскрашенном графе и т. п. Применяются и другие способы упорядочения, однако все они в общем случае не обеспечивают раскраску графа в минимальное число цветов χ .

В [2] для решения одной комбинаторной задачи распределения ограниченного ресурса между объектами также используется последовательный алгоритм, при этом по результатам каждого этапа его применения объекты, которым не удалось выделить ресурс, перемещаются в начало списка. Воспользуемся этой идеей для решения задачи раскраски графа. Для этого положим степени трудности присвоения d_i (будем называть их весами) каждой вершины v_i одинаковыми, произвольно упорядочим вершины и, применив последовательный алгоритм, найдем допустимую раскраску. Пусть при этом использовано q_k цветов. Увеличим веса d_i всех тех вершин, которым присвоен цвет с максимальным номером q_k , на случайную величину

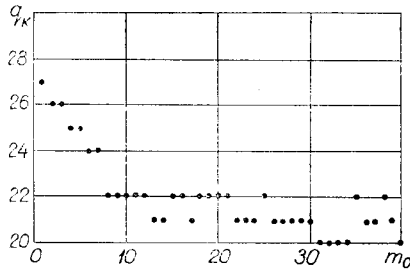


Рис. 1

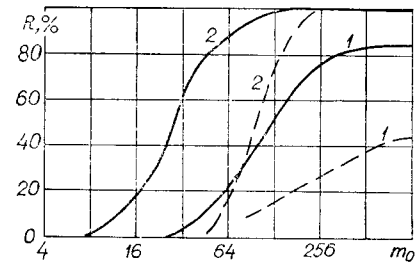


Рис. 2

Δd_i и упорядочим вершины в порядке убывания d_i (использование в качестве Δd_i случайной величины введено для исключения заикливания). При этом трудные для присвоения вершины окажутся в начале списка. Вновь воспользуемся последовательным алгоритмом, причем после каждого его применения веса вершин, которым при раскраске понадобятся цвета с максимальными номерами, опять увеличим и снова упорядочим список вершин. Таким образом, в процессе работы алгоритма происходит самообучение, в результате которого список вершин итерационно перепорядочивается. Эти соображения приводят к следующей процедуре раскраски графа.

1. Произвольно упорядочить вершины графа и окрасить их с помощью последовательного алгоритма. Запомнить раскраску и определить потребовавшееся число цветов q_k . Поставить в соответствие каждой вершине v_i вес d_i и положить все $d_i := 1$.
2. Положить $m := 1$ и $q_k := q_k - 1$.
3. С помощью последовательного алгоритма окрасить вершины графа.
4. Если ни одной вершине графа не понадобился цвет, номер которого больше q_k , запомнить раскраску и перейти к шагу 2.
5. Положить $m := m + 1$. Если $m > m_{\max}$, перейти к шагу 7.
6. Для каждой вершины v_j , которой понадобился цвет с номером, большим q_k , увеличить вес d_j на случайную величину Δd_j и упорядочить вершины графа в порядке неубывания d_j . Перейти к шагу 3.
7. Положить $q_k := q_k + 1$ и закончить работу.

Алгоритм заканчивается, когда число безуспешных попыток (циклов) окраски графа в $q_k - 1$ цветов окажется равным заданному числу m_{\max} . При этом на выходе будем иметь раскраску графа в q_k цветов.

Для проверки эффективности алгоритма в соответствии с достаточно широко известной процедурой генерировались графы, с большой вероятностью обладающие заданным хроматическим числом χ . Суть процедуры генерации заключается в следующем. С помощью датчика случайных чисел, равномерно распределенных в диапазоне $[0, 1]$, генерируются прямоугольные координаты точки, являющейся первой вершиной графа, и ей присваивается первый цвет. Генерируются координаты следующей точки v_i — кандидата на включение в число вершин графа. Она соединяется дугами со всеми теми уже включенными в граф точками v_j , расстояния от нее до которых $r_{ij} \leq r_{\max}$, где r_{\max} — заранее заданная величина. Если теперь на смежных с v_i вершинах не встречается цвет c_j с номером $j \leq \chi$, он присваивается этой вершине, вершина включается в граф и генерируется следующая точка. Процедура прекращается после того, как в достаточно длинной серии ни одна попытка генерации точек-кандидатов не приводит к увеличению числа вершин в графе. Практически, например, при $\chi = 20$ и $r_{\max} = 0,3$ при числе попыток $N \geq 1000$ получается граф, содержащий 201 ± 4 вершины, а при $r_{\max} = 0,6$ — 71 ± 2 вершины.

Изображенный на рис. 1 график демонстрирует работу алгоритма с самообучением при раскраске одного из графов с $\chi = 20$ и числом вершин $n = 71$. В качестве Δd_j на шаге 6 использовались равномерно распределенные в диапазоне $[0, 1]$ случайные числа. По оси абсцисс графика отложено общее число обращений m_0 к последовательному алгоритму, а по оси ординат — потребовавшееся для раскраски число цветов q_k . В приведенном примере при $m_0 = 31$ было найдено глобально оптимальное решение.

Для оценки эффективности алгоритма раскраски с самообучением он был применен к 256 графам с $\chi = 20$, $n = 71 \pm 2$ и к 18 графам с $\chi = 20$, $n = 201 \pm 4$. Исходная нумерация вершин задавалась случайной с равномерным распределением, а $m_{\max} = 1000$. При этом для 214 графов (83,6 %) с $n = 71 \pm 2$ была найдена раскраска точно в χ цветов, а для остальных 42 графов (16,4 %) — в $\chi + 1$ цвет. Для графов с числом вершин 201 ± 4 глобально оптимальное решение получено в 43,8 % случаев, а в остальных 56,2 % получена раскраска в $\chi + 1$ цвет.

На рис. 2 представлена зависимость относительного числа графов R , для которых найдена глобально оптимальная раскраска, от числа обращений к последовательному алгоритму (кривые 1). Кривые 2 отображают относительное число графов, для которых найдена раскраска не хуже, чем в $\chi + 1$ цвет. При этом сплошные линии соответствуют графам с $n = 71 \pm 2$, а штриховые — графам с $n = 201 \pm 4$. На рис. 3 показано среднее число обращений $m_{\text{ср}}$ к последовательному алгоритму для

получения раскраски указанных графов в q_k цветов. Процессорное время, понадобившееся в среднем при $m_0 = 1000$ для решения задачи раскраски одного графа на ЭВМ ЕС 1045 при $n = 71 \pm 2$, составило 31,6 с, а при $n = 201 \pm 4 - 300$ с, т. е. увеличение числа вершин в 2,8 раза привело к увеличению требуемого времени в 9,5 раз. При этом основная часть времени уходит на поиск раскраски в χ цветов. Для сравнения среднее время получения раскраски в $\chi + 1$ цвет для $n = 71$ составило 3,8 с, а для $n = 201 - 35,4$.

Сопоставление этих данных с рис. 3 свидетельствует о том, что затраты машинного времени связаны в основном с работой последовательного алгоритма, временная сложность которого оценивается величиной $O(n^2)$.

Приведенные примеры демонстрируют эффективность применения предлагаемого алгоритма к раскраске графов (по крайней мере, рассматриваемого вида) с числом вершин, достигающим нескольких сотен. Представляется, что алгоритм может оказаться полезным и в случае графов с существенно большим числом вершин, если перед его применением использовать алгоритм Ершова — Кожухина, приводящий исходный граф к графу с меньшим числом вершин [1].

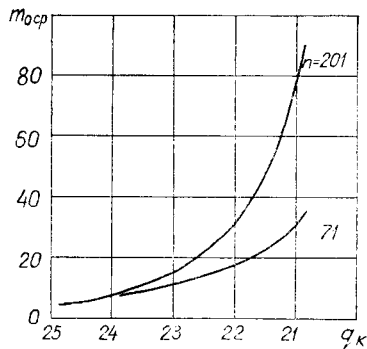


Рис. 3

СПИСОК ЛИТЕРАТУРЫ

1. Евстигнеев В. А. Применение теории графов в программировании. — М.: Наука, 1985.
2. Vox F. A heuristic technique for assigning frequencies to mobile radio nets // IEEE Trans. on Vehicular Techn.— 1978.— VT-27, N 2.— P. 57.

Поступило в редакцию 11 ноября 1988 г.

УДК 621.391.837

В. И. ГУПАЛ, И. В. ТРОХИМЕЦ
(Киев)

НЕКОТОРЫЕ ОСОБЕННОСТИ ОБРАБОТКИ ИЗОБРАЖЕНИЙ В ПЕРВИЧНЫХ ОТДЕЛАХ ЗРИТЕЛЬНОЙ СИСТЕМЫ ЧЕЛОВЕКА

При проектировании различных систем обработки изображений, конечным звеном которых является человек, большое внимание уделяется вопросам, связанным со свойствами его зрительной системы. В настоящее время подход, основанный только на учете статистических характеристик изображений, достиг предела своих возможностей и успехов можно ожидать лишь при условии согласования системы обработки изображений с особенностями зрительного анализатора человека. Это обстоятельство наглядно проявляется при разработке систем эффективного кодирования изображений [1]. С другой стороны, знание свойств зрительного анализатора человека необходимо для создания автоматических устройств распознавания изображений [2]. В этой связи большой интерес представляет исследование механизмов анализа изображений, протекающих в зрительной системе человека [1—5]. Одним из удобных путей анализа этих механизмов является исследование положений точек фиксации взора при осмотре изображений, так как анализ в зрительной системе тесно связан с движениями глаз при осмотре. На основе изучения движений глаз при осмотре в [2] сформулирована и экспериментально обоснована гипотеза, согласно которой локальные геометрические особенности изображений (информативные для человека фрагменты) могут быть выделены с помощью предложенной авторами специальной поисковой функции — функции информативности (ФИ). Значение этой функции в каждой точке изображения (в центре рассматриваемого фрагмента) есть число, характеризующее степень «регулярности» изображения в пределах данного фрагмента. Для определения этого числа в [2] использовалось стандартное изображение с максимальной нерегулярностью (например, черное пятно на белом фоне). Тогда значение функции информативности представляет собой меру различия рассматриваемого и стандартного фрагментов, понимаемую как евклидово расстояние между этими фрагментами. Предложенный таким образом подход на основе анализа экстремумов ФИ объяснил с единых позиций множество иллюзий, присущих человеческому зрению, и позволил определить, какие фрагменты изображений наиболее информативны для зрительной системы человека. Однако открытым остался вопрос, почему точки фиксации взора совпадают с экстремумами введенной таким искусственным путем ФИ. Прежде чем ответить