ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Октябрь—декабрь 1995. Том 2, № 4, 74–79

УДК 519.1

НЕКОТОРЫЕ ЗАДАЧИ ОПТИМИЗАЦИИ РАСПИСАНИЯ ПЕРЕДАЧИ СООБЩЕНИЙ В ЛОКАЛЬНОЙ СЕТИ СВЯЗИ

А. В. Пяткин

Задача построения оптимального расписания передачи информации в локальной сети связи сводится к задаче нахождения *смешанной* раскраски дуг в ориентированном мультиграфе минимальным числом цветов. Приводятся два алгоритма для решения последней задачи. Термин *смешанная раскраска* означает, что некоторые дуги могут быть разбиты на две части и раскрашены в два цвета, причем так, чтобы номер цвета первой части не превосходил номера цвета второй части.

1. Содержательная постановка задачи

Пусть задана локальная сеть, состоящая из центральной ЭВМ и n шин, соединяющих ее с периферийными объектами. Эти объекты могут передавать информацию друг другу. Если оба объекта находятся на одной шине, то их «общение» происходит без участия центральной ЭВМ. В статье такой вид общения не рассматривается. Объекты, находящиеся на разных шинах, могут общаться только через центральную ЭВМ. Суммарное количество информации, которую все объекты i-й шины должны передать объектам j-й шины, обозначается через d_{ij}. (Из сказанного выше следует, что $d_{ii} = 0$ для любого i.) Считается, что пропускные способности всех шин одинаковы и равны c единиц информации в единицу времени. Тогда время t_{ij} передачи d_{ij} единиц информации из i-й шины в j-ю определяется по формуле

$$t_{ij} = \frac{d_{ij}}{c}.$$

Будем считать, что все t_{ij} являются целыми неотрицательными числами. «Общение» шин между собой может быть организовано следующими способами:

1) центральная ЭВМ соединяет i-ю шину с j-й напрямую. В этом случае информация из i-го объекта передается в j-й (или, наоборот, из j-го в i-й) непосредственно, т. е. без запоминания в центральной ЭВМ;

© 1995 Пяткин А. В.

2) информацию, которую i-я шина должна передать j-й шине, можно сначала получить по i-й шине и запомнить в центральной ЭВМ, а затем передать по j-й шине адресату.

Ограничение. Каждая шина в любой момент времени может общаться не более чем с одним объектом (под объектом понимаются как периферийные пользователи, так и центральная ЭВМ). Это ограничение не распространяется на центральную ЭВМ, которая может получать или передавать информацию одновременно нескольким шинам (при втором способе общения).

Весь процесс общения представляет собой сеанс, т. е. передаче подлежит лишь тот объем информации, который был «заявлен» до начала общения (другими словами, d_{ij} не изменяются во время общения). Требуется так составить расписание передачи информации между шинами, чтобы общая продолжительность сеанса была наименьшей.

2. Построение математических моделей

Предположим, что второй способ общения запрещен, т. е. шины могут быть связаны только напрямую. В этом случае независимо от того, должен ли объект, находящийся на i-й шине, передать информацию объекту, находящемуся на j-й шине (или наоборот), необходимо связать i-ю и j-ю шины. Таким образом, можно ввести понятие «потребность в общении между i-й и j-й шинами»

$$p_{ij} \stackrel{\text{def}}{=} t_{ij} + t_{ji}.$$

Рассмотрим n-вершинный неориентированный мультиграф $G = (V, E)$ с множеством вершин V и множеством ребер E такой, что i-й шине соответствует i-я вершина графа, а число ребер, соединяющих i-ю и j-ю вершины, равно p_{ij}.

Составление расписания общения шин в этом случае означает, что каждому ребру $e \in E$ должен быть поставлен в соответствие момент времени $f(e) \in \{1, \ldots, T\}$, когда данная потребность в общении будет реализована (T — общая продолжительность сеанса); при этом для любых ребер e_1, e_2, инцидентных одной вершине, должно выполняться неравенство $f(e_1) \neq f(e_2)$. Требуется минимизировать T.

Ясно, что если в написанном выше тексте слова «момент времени» заменить на слово «цвет», то получим классическую задачу реберной раскраски мультиграфа (искомое T есть хроматический индекс мультиграфа G). Эта задача NP-трудна [1]. Известны лишь приближенные алгоритмы ее решения, в частности алгоритм Визинга [2–4].

Вернемся к исходной задаче, когда возможны как первый, так и второй способ передачи сообщений. Второй способ передачи информации от объекта с i-й шины к объекту с j-й шины, как правило, осуществляется, если в данный момент времени j-я шина занята, а i-я — свободна. Тогда информация по i-й шине передается в центральную ЭВМ, а впоследствии, когда j-я шина освободится, передается объекту с j-й шины. Таким образом, становится существенной не только «потребность в общении», но и *направление* передачи информации.

Поэтому рассмотрим *ориентированный* мультиграф $G = (V, E)$ с n вершинами, в котором число дуг, ведущих из i-й вершины в j-ю вершину, равно $t_{ij}, 1 \leqslant i, j \leqslant n$.

От задачи построения расписания к задаче построения раскраски дуг перейдем следующим образом: пусть $e = (i, j) \in E$ — дуга графа, а $f(e)$ — ее цвет. Будем считать, что $f(e) = a$, если в момент времени a шины i и j связаны напрямую, и $f(e) = (a, b)$, если в момент времени a информация от i-го объекта передается в центральную ЭВМ, а в момент времени b — из центральной ЭВМ j-му объекту. Обозначение $f(e) = (a, b)$ следует понимать так: дуга e разбита на две части и та ее часть, которая инцидентна вершине i, окрашена в цвет a, а другая — в цвет b. Такую раскраску будем называть *смешанной*.

Здесь необходимо учитывать следующее. Чтобы в момент времени b передать информацию из центральной ЭВМ объекту с j-й шины, необходимо перед этим загрузить ее в центральную ЭВМ по i-й шине. Таким образом, должно быть выполнено следующее условие допустимости: если $e \in E$ и $f(e) = (a, b)$, то $a < b$.

Смешанную раскраску, удовлетворяющую этому условию, будем называть *допустимой*.

Через s обозначим максимальную степень графа, а через p — максимальное количество ребер в мультиребре, т. е.

$$s = \max_i \sum_{j=1}^{n} (t_{ij} + t_{ji}),$$

$$p = \max_{i,j} (t_{ij} + t_{ji}).$$

Очевидно, что для любой правильной раскраски дуг (т. е. такой раскраски, когда каждой вершине инцидентно не более одной дуги каждого цвета) требуется не менее s цветов. Алгоритм Визинга позволяет правильно раскрасить дуги мультиграфа не более чем $s + p$ цветами, не используя смешанно окрашенных дуг. Легко построить смешанную, правильную, но возможно недопустимую раскраску дуг мультиграфа в s цветов. Действительно, разбиваем каждую дугу на две части и в полученном двудольном графе правильно окрашиваем дуги. По теореме Кенига [5] для этого достаточно s цветов.

В настоящей статье устанавливается следующий факт.

Теорема. *В любом ориентированном мультиграфе G степени s существует правильная смешанная допустимая раскраска его дуг в s цветов.*

Поиски конструктивного доказательства этой теоремы (т. е. алгоритма построения такой раскраски) велись по двум принципиально разным направлениям:

1) сначала строится правильная смешанная частичная раскраска дуг графа, затем число окрашенных дуг увеличивается вследствие перекраски уже окрашенных дуг;

2) сначала строится правильная смешанная, но недопустимая раскраска графа в s цветов, затем с помощью некоторых преобразований, не увеличивающих число цветов, имеющаяся раскраска превращается в допустимую.

Предложенные алгоритмы **А1** и **А2** доказывают теорему.

3. Алгоритм А1

Пусть G — ориентированный мультиграф степени s, дуги которого частично окрашены не более чем в s цветов. Для произвольной его вершины A обозначим через $M(A)$ множество цветов из набора $\{1, \ldots, s\}$, не использованных при окраске дуг, инцидентных вершине A. Пусть $f(AB)$ — цвет дуги, ведущей из A в B. Все неокрашенные дуги будем называть *выделенными*. Предлагается алгоритм, позволяющий раскрасить выделенные дуги, сохраняя правильность и допустимость получающейся смешанной раскраски, т. е. строящий правильную допустимую смешанную раскраску дуг мультиграфа G в s цветов.

Шаг алгоритма заключается в анализе одной из выделенных дуг и выполнении одной из описанных ниже процедур.

Пусть AB — выделенная дуга. Ясно, что $M(A) \neq \varnothing$ и $M(B) \neq \varnothing$.

I. Если $M(A) \cap M(B) \neq \varnothing$, то дуга AB красится в цвет a, где $a \in (M(A) \cap M(B))$. При этом правильность раскраски не нарушается.

II. Если $M(A) \cap M(B) = \varnothing$, то находятся минимальный цвет a из $M(A)$ и максимальный цвет b из $M(B)$. Если $a < b$, то полагаем $f(AB) = (a, b)$, и полученная раскраска дуг остается допустимой.

III. В случае $a > b$ строится цветочередующаяся (a, b)-цепь с началом в вершине B (эта цепь не пуста, так как $a \notin M(B)$). Предположим, что в процессе построения цепи ни разу не встретилась смешанно окрашенная дуга. В этом случае цепь конечна, так как она не может самопересекаться, т. е. ее длина не превосходит числа вершин графа. Если цепь кончается в вершине, отличной от A, то дуги цвета a в рассматриваемой цепи перекрашиваются в цвет b, а дуги цвета b — в цвет a, что не нарушает правильности раскраски, и полагается $f(AB) = a$. Так как нет дуг цвета a, инцидентных вершине A (по определению $M(A)$), а дуга цвета a, инцидентная вершине B, перекрашена в цвет b, то правильность раскраски при перекраске дуги AB не нарушается.

IV. Если цепь кончается в вершине A, то осуществляется перекраска $a \leftrightarrow b$ и полагается $f(AB) = (b, a)$.

V. Допустим, что при построении цепи в случае $a > b$ встретилась смешанно окрашенная дуга e. Тогда осуществляется перекраска $a \leftrightarrow b$ всех дуг цепи, кроме дуги e, полагается $f(AB) = a$ и выделяется дуга e (т. е. дуга e становится неокрашенной). Алгоритм заканчивает работу, когда не останется ни одной выделенной дуги.

Замечание. Легко проверить, что в процессе раскраски и перекраски дуг правильность и допустимость окраски графа не нарушается. Покажем, что алгоритм **A1** конечен, т. е. не зацикливается. Процедуры **I** и **III** уменьшают число неокрашенных дуг на 1; процедуры **II** и **IV** также уменьшают число неокращенных дуг на 1, при этом увеличивая на 1 число смешанных дуг. Вначале смешанных дуг нет — они могут появиться лишь в результате выполнения процедур **II** или **IV**. Процедура **V** уменьшает на 1 число смешанных дуг, сохраняя число выделенных дуг неизменным. Так как процедура **V** выполняется лишь при наличии в графе по крайней мере одной смешанной дуги, то даже если на некотором шаге алгоритма имеется m смешанных дуг, не позднее чем через m шагов придется выполнить одну из процедур **I–IV**, что приведет к уменьшению числа неокрашенных дуг. Таким образом, алгоритм конечен. В наихудшем случае его работа закончится через $2N - 1$ шагов, где N — число дуг мультиграфа. При этом используется не более $O(n^4p^2)$ элементарных операций.

4. Алгоритм A2

Пусть ориентированный мультиграф G с n вершинами имеет правильную смешанную, но недопустимую раскраску в s цветов. Выделяем все недопустимо раскрашенные дуги.

Шаг алгоритма заключается в анализе некоторой выделенной дуги и выполнении одной из следующих процедур.

I. Пусть AB — выделенная дуга, $f(AB) = (a, b)$. Так как дуга AB недопустима, то $a > b$. Строится цветочередующаяся (a, b)-цепь с началом в вершине B. Пусть эта цепь не содержит смешанных дуг. Тогда в ней не может быть более чем n звеньев, т. е. цепь конечна (в частности, она может быть пустой). Если цепь оканчивается не в вершине A, то осуществляется перекраска $a \leftrightarrow b$ и полагается $f\ (AB)\ =\ a$.

II. Если цепь кончается в A, то осуществляется перекраска $a \leftrightarrow b$ и полагается $f(AB) = (b, a)$. (Используется условие $a > b$.)

III. Предположим, что в процессе построения цепи встретилась смешанная дуга е. Тогда $f(e) = (a, c)$ или $f(e) = (b, c)$. Рассмотрим случай $f(e) = (a, c)$. Тогда $c \neq a$, так как дуга e — смешанная. Осуществляется перекраска $a \leftrightarrow b$ и полагается $f(AB) = a$, $f(e) = (b, c)$. При этом дуга e может превратиться из допустимой в недопустимую (или наоборот). В этом случае ее нужно внести в «список» выделенных дуг (или вычеркнуть).

Случай $f(e) = (b, c)$, $c \neq b$, рассматривается аналогично.

Алгоритм заканчивает работу, когда не останется выделенных дуг.

Покажем, что алгоритм **A2** конечен. Пусть k — число выделенных дуг, а N — общее число смешанных дуг перед началом работы алгоритма. Можно считать, что $k \leqslant N/2$, а $N \leqslant n(n-1)p/2$.

Процедура **I** превращает одну недопустимую дугу в обычную, процедура **II** — одну недопустимую дугу в допустимую, а процедура **III** — одну смешанную дугу в обычную. Таким образом, число шагов алгоритма не превосходит $k + N$, т. е. не превосходит $3n(n-1)p/4$. При этом используется не более чем $O(n^4p^2)$ элементарных операций.

ЛИТЕРАТУРА

1. **Holyer I.** The NP-completeness of edge-coloring // SIAM J. Comput. 1981. V. 10, N 4. P. 718–720.
2. **Визинг В. Г.** Об оценке хроматического класса p-графа // Дискретный анализ: Сб. науч. тр. Новосибирск: Ин-т математики СО АН СССР, 1964. Вып. 3. С. 25–30.
3. **Визинг В. Г.** Критические графы с данным хроматическим классом // Дискретный анализ: Сб. науч. тр. Новосибирск: Ин-т математики СО АН СССР, 1965. Вып. 5. С. 9–17.
4. **Визинг В. Г.** Хроматический класс мультиграфов // Кибернетика. 1965. № 3. С. 29–39.
5. **Зыков А. А.** Основы теории графов. М.: Наука, 1987.

Адрес автора:

Россия,
630090 Новосибирск,
ул. Пирогова, 2,
Новосибирский
государственный университет

Статья поступила

5 декабря 1994 г.