О ДВУХ ЗАДАЧАХ ТРАССИРОВКИ

Искандар Карапетян
ИПИА НАН РА
Ереван, Армения
e-mail: isko@ipia.sci.am

Самвел Дарбинян
ИПИА НАН РА
Ереван, Армения
samdarbin@ipia.sci.am

Резюме

В работе рассматриваются две задачи, возникающие на этапе трассировки, при проектировании БИС и СБИС [1, 2, 3, 4]. 1) Задано семейство интервалов $F=\{f_1, f_2,\dots, f_n\}$, где $f_i=[a_i, b_i]$, $i=\overline{1,n}$ и множество цветов $I=\{1,2,\dots,m\}$. Предположим, что для каждого интервала f_i, $i=\overline{1,n}$, заранее выделен некий интервал $I(i)$ из множества I, лишь цветами которого можно окрасить интервал f_i. Требуется найти наибольшее подсемейство семейства F, имеющее правильную раскраску в рамках указанных ограничений, и ее соответствующую правильную раскраску. 2) На прямоугольной решетке $L\times K$ задано множество отрезков $\{l_1, l_2,\dots, l_m\}$. Требуется найти путь наименьшей длины, проходящей через все заданные отрезки .

Ключевые слова
Трассировка, интервал, правильная раскраска, решетка, поток в сети.

Введение

В связи с канальной трассировки, впервые задача γ-раскраски(заданными γ цветами правильно раскрасить наибольшее число вершин графа) вершин граф интервалов была поставлена и эффективно решена С. Маркосяном [5]. Позже задача γ-раскраски была эффективно решена для триангулированных [6], транзитивно орентируемых [7] и других классов графов. Ясно, что выше упомянутая задача в общем случае является NP-полным. Первая рассматриваемая задача возникает на этапе глобальной трассировки, при проектировании матричных БИС и формулируется следующим оброзом:

1) Задано семейство интервалов $F=\{f_1, f_2,\dots, f_n\}$, где $f_i=[a_i, b_i]$, $i=\overline{1,n}$ и множество цветов $I=\{1,2,\dots,m\}$ (в качестве цветов берется начальный отрезок натурального ряда). Предположим, что для каждого интервала f_i, $i=\overline{1,n}$, заранее выделен некий интервал $I(i)$ из множества I, лишь цветами которого можно окрасить интервал f_i. Требуется найти наибольшее подсемейство семейства F, имеющее правильную раскраску в рамках указанных ограничений, и ее соответствующую правильную раскраску.

Очевидно,что первая задача является обобщением задачи γ-раскраски граф интервалов. Г. Гаспарян (устное сообщение) доказал, что первая задача является NP-полным. Доказательство, на самом деле, очень простое. Открывая окружность L в некотором точке $x\in L$ и выпрямляя его, каждая дуга, содержащая точку x, превращается в две интервалы, а остальные дуги превращаются в интервалы. Следовательно задача раскраски дуг окружности сводится к раскраске полученных интервалов, так, чтобы две интервалы каждой дуги были окрашены одним и тем же цветом. Поскольку задача минимальной раскраски дуг окружности NP-полна [8], то первая задача NP-полна. Вторая задача возникает на этапе детальной трассировки, при трассировки специальных цепей, и формулируется так:

2) На прямоугольной решетке $L\times K$ задано множество отрезков $\{l_1, l_2,\dots, l_m\}$. Требуется найти путь наименьшей длины, проходящей через все заданные отрезки.

В работе предложены приближенные алгоритмы решения сформулированных задач, которые в отдельных случаях являются точными.

Основные результаты

При проектировании БИС и СБИС задачи, возникающие на этапе трассировки, как правило, являются NP- полными. В работе рассматриваются две такие задачи.

Задача1.Задано семейство интервалов $F=\{f_1, f_2,\ldots, f_n\}$, где $f_i=[a_i, b_i], i=\overline{1,n}$, и множество цветов $I=\{1,2,\ldots,m\}$. Предположим, что для каждого интервала $f_i, i=\overline{1,n}$, заранее выделен некий интервал $I(i)$ из множества I, лишь цветами которого можно окрасить интервал f_i. Требуется найти наибольшее подсемейство семейства F, имеющее правильную раскраску в рамках указанных ограничений, и ее соответствующую правильную раскраску. Здесь в качестве цветов берется начльный отрезок натурального ряда.

Описание алгоритма решения задачи 1.

Шаг 0. Упорядочим интервалы семейства F по возрастанию правых концов. Чтобы не вводить новые обозначения, мы можем предположить, что первоначальное упорядочение является таковим т. е. $b_1 \le b_2 \le \ldots \le b_n$. Отметим, что алгоритм раскрашивает интервалы в обратном порядке упорядочения.

Шаг 1. Окрасим интервал f_n одним из цветов множества $I(n)$. Все остальные цвета множества $I(n)$ будем считать резервными для интервала f_n и обозначим множество этих цветов через $I_p(n)$.

Шаг i. На i-ом шаге рассмотривается интервал f_{n-i+1}. Если во множестве $I(n-i+1)$ есть цвета, которыми можно правильно окрасить интервал f_{n-i+1}, то одним из них, скажем c_i-ом цветом, окрасим f_{n-i+1}, а остальные цвета будем считать резервными для f_{n-i+1} и обозначим множество этих цветов через $I_p(n-i+1)$. Для каждого окрашенного интервала f_j, который пересекается с f_{n-i+1}, удаляется из $I_p(j)$ цвет c_i. Если ни одним из цветов множества $I(n-i+1)$ нельзя правильно окрасить интервал f_{n-i+1}, то поступаем следующим образом: если для некоторого цвета $\nu \in I(n-i+1)$ существует окрашенный интервал (ν-цветом), который пересекается с f_{n-i+1}, то перекрасим его в наименьшим резервным цветом, а интервал f_{n-i+1} окрасим в цвет ν. А множество резервных цветов окрашенных интервалов преобразуется соответствующим образом, как было сказано выше. В противном случае рассматриваются все окрашенные интервалы $f_{j_1}, f_{j_2}, \ldots, f_{j_k}$, цвета которых принадлежат $I(n-i+1)$ и которые пересекаются с интервалом f_{n-i+1}. Если $a_{n-i+1} \ge \min\{a_{j_1}, a_{j_2}, \ldots, a_{j_k}\}$, то считаем, что f_{n-i+1} не окрашен (удаляется) и перейдем к шагу $i+1$. Иначе, окрасим интервал f_{n-i+1} цветом того окрашенного интервала, среди интервалов $f_{j_1}, f_{j_2}, \ldots, f_{j_k}$, левый конец которого является наименьшим. Считаем, что последний интервал не окрашен (удаляется) и перейдем к шагу $i+1$.

Теорема 1. Если имеет место $I(1)=I(2)=\ldots=I(n)=\{1,2,\ldots,k\}$,то описанный алгоритм является точным.

Доказательство. По индукции докажем, что на каждом шаге алгоритм раскрашивает наибольшее число интервалов среди рассматриваемых. Если $i \le \kappa$, то очевидно, что алгоритм раскрашивает все $f_n, f_{n-1}, \ldots, f_{n-k+1}$ интервалы. Предположим, что при $i=l$, алгоритм раскрашивает наибольшее число интервалов среди интервалов $f_n, f_{n-1}, \ldots, f_{n-l+1}$. Докажем, что когда $i=l+1$ алгоритм тоже раскрашивает наибольшее число интервалов среди интервалов $f_n, f_{n-1}, \ldots, f_{n-l}$. Если алгоритм раскрашивает интервал f_{n-l} и, притом, ни один из окрашенных интервалов не удаляется, то нечего доказать. В противном случае существуют k окрашенных интервалов, которые содержат правый конец интервала f_{n-l} т.е. b_{n-l}. Поскольку k цветами невозможно правильно раскрасить $k+1$ попарно пересекающиеся интервалы, следовательно один из них не раскрашивается, что и делает алгоритм. Теорема доказана.

Следующая теорема показывает, что первая задача полиномиално разрешима, при следующих ограничениях.

Подсемейство всех интервалов из F, которые можно окрасить i-ым цветом, обозначим через F_i, а множество всех различных клик подсемейства F_i-через $\{Q_{i,1}, Q_{i,2}, \dots, Q_{i,k(i)}\}$, $i = \overline{1,m}$. Скажем, что семейство интервалов F удовлетворяет условию P, если пересечение любых двух различных клик одного и того же подсемейства F_i пусто, т. е. для каждого $i, i = \overline{1,m}$ и для всех $j_1, j_2, j_1 \neq j_2$ и $j_1, j_2 = \overline{1, k(i)}$ имеет место $Q_{i,j_1} \cap Q_{i,j_2} = \varnothing$.

<u>Теорема</u> 2. Задача 1, при выполнении условия P, полиномиально разрешима.

Доказательство. Построим сеть $G = (X, E)$ следующим образом. Чтобы не вводить новые обозначения просто, в качестве множества вершин X сети G, возьмем все интервалы семейства F, все клики подсемейства F_i, $i = \overline{1,m}$, все цвета множества I и еще две другие вершины s и t, которые назовем, соответственно, источником и стоком. А множество дуг E сети G строим так: источник s соединяем со всеми интервалами F с помощью дуг, исходящих из s. Сток t соединяем со всеми цветами с помощью дуг, входящих в t. Все интервалы каждой клики $Q_{i,j}$ соединяем с помощью дуг, входящих в $Q_{i,j}$, и кроме того, $Q_{i,j}$ соединяем с соответствующей i-ой цветом с помощью дуги, исходящей из клики $Q_{i,j}$. Будем считать, что пропускные способности всех дуг, кроме тех, которые входят в сток t, равны 1, а пропускные способности последних равны n.

Имея потоки дуг некоторого максимального потока сети G, можно найти некоторое наибольшее подсемейство F_0 из F, которое имеет правильную раскраску при заданных ограничениях и саму раскраску. Для этого необходимо взять все те интервалы, протекающий поток по которым в сети G равен 1. Цвет выбранных интервалов однозначно определяется единственным насыщенным путем, началом которого является интервал, а конец- цвет, которым окрашен этот интервал.

То, что одноцветные интервалы не пересекаются, обеспечено тем, что из каждой клики выходит только одна дуга и то, что все клики одной и той же F_i попарно непересекаются. А максимальность подсемейства F_0 обеспечена выбором максимального потока в сети. Теперь, так как число вершин сети G не превосходит $nm+n+m+2$ и задача нахождения максимального рационального потока полиномиально разрешима, то задача 1, при условии P, полиномиально разрешима. Теорема доказана.

Теперь перейдем к формулировке второй задачи.

<u>Задача</u> 2. На прямоугольной решетке $L \times K$ задано множество отрезков $\{l_1, l_2, \dots, l_m\}$,не содержащие три отрезка, проходящие через одну вершину решетки.Требуется найти путь наименьшей длины, проходящей через все заданные отрезки.

Ниже предложен приближенный алгоритм решения этой задачи, состающий из двух этапов.

<u>Первый этап</u>. Построим полный взвешенный граф $G = (X, E)$ следующим образом. В качестве множества вершин графа G берем все концы всех отрезков $l_1, l_2, \dots, l_m$. При том, если вершина решетки является концом двух отрезков, то в графе G этому концу соответствуют две вершины. А длина ребра графа G считается расстояние между соответствующими вершинами решетки. Затем из графа G удаляем все ребра, которые соответствуют заданным

отрезкам. Нетрудно показать, что полученный граф имеет совершенное паросочетание. С помощью венгерского алгоритма [9] находим наименьшее совершенное паросочетание. Пусть это будет $u_{i_1,j_1}, u_{i_2,j_2}, \ldots, u_{i_m,j_m}$. Рассмотрим граф G', порожденный множеством ребер найденного паросочетания и соответствующими ребрами заданных отрезков. Граф G', в общем случае, может состоять из нескольких циклов, скажем $C_1, C_2, \ldots, C_k$. Обозначим через $X_i = \{x_1^i, x_2^i, \ldots\}$ множество вершин цикла $C_i, i = \overline{1,k}$, а через d_{ij} расстояние между множествами X_i и X_j.

Второй этап. Пусть $d_{i_0 j_0}$ - есть минимальное расстояние среди всех $d_{ij}, i \neq j,\ i,j = \overline{1,k}$. Предположим, что $d_{i_0 j_0}$ достигается на вершинах $x_\mu^{i_0} \in X_{i_0}$ и $x_\nu^{j_0} \in X_{j_0}$. Построим путь, проходящий через все вершины циклов C_{i_0} и C_{J_0}. Для этого удалим те ребра паросочетания, которые инцидентны вершинам $x_\mu^{i_0}$ и $x_\nu^{j_0}$. Остальные ребра циклов C_{i_0} и C_{j_0} вместе с ребром графа G', обеспечивающего минимальное расстояние между множествами X_{i_0} и X_{j_0}, составят требуемый путь. Поскольку описание следующих шагов алгоритма одинаково, то опишем очередной шаг алгоритма. Обозначим через L - путь, найденный на предыдущем шаге, а его концы через x' и x''. На очередном шаге вычесляется расстояние x' и x'' от всех остальных циклов и среди них берется цикл, минимально отдаленный от x' или от x''. Найденное расстояние, в качестве ребра, и соответствующий цикл добавляется к пути L. А из добавленного цикла до этого удаляется соответствующее ребро паросочетания. Этот процесс продолжается до тех пор, пока не найден путь, проходящий через все заданные отрезки.

Замечание 1. Если в результате первого этапа получаем только один цикл, то удаляя самое длинное ребро паросочетания, получаем оптимальное решение задачи 2.

Замечание 2. Выше описанный алгоритм успешно работает и на плоскости и в простанстве.

Литература

[1]. T. Ohtsuki (Editor). Layout Design and Verification. North Holland, 1986.

[2]. N.Sherwani. Algorithms for VLSI Physical Design Automation. Kluwer Academic Publishers, 1995.

[3]. Дж. Д. Ульман. Вычислительные Аспекты СБИС. Москва, „Радио и связь” , 1990.

[4]. М. Ватанабэ, К. Асада, К. Кани, Т. Оцуки. Проектирование СБИС. Москва, „Мир”,1988.

[5]. А. В. Петросян, С.Е. Маркосян, Ю. Г. Шукурян. Математические Вопросы Автоматизации Проектирования ЭВМ. Ереван, 1977.

[6]. M. Yannakakis, F. Gavril. The Maximum $k-$colorable Subgraph Problem for Chordal Graphs. Information Processing Letters. 24, N2, pp.133-137,1978.

[7]. F. Gavril. Algorithms for Maximum k-colorings and k-coverings of Transitive Graphs. Networks, vol.,17, N4, pp.465-470,1987.

[8]. M. R. Garey, D. S. Johnson, G. L. Miller, and G. H. Papadimitriou. The Complexity of coloring circular arcs and chords. SIAM Journal on Algebraic and Discrete Methods. 1(2) pp. 216 – 227, June, 1980.

[9]. Х.Пападимитриу, К. Стайглиц, Комбинаторная оптимизация. Москва, „Мир”,1985.