Р О С С И Й С К А Я А К А Д Е М И Я Н А У К
СПЕЦИАЛЬНАЯ АСТРОФИЗИЧЕСКАЯ ОБСЕРВАТОРИЯ

П Р Е П Р И Н Т N 199

Д.А.Павлов, Р.А.Кочкаров

АЛГОРИТМЫ С ОЦЕНКАМИ ПОСТРОЕНИЯ ПОКРЫТИЙ НЕПЕРЕСЕКАЮЩИМИСЯ ПРОСТЫМИ ЦЕПЯМИ НА ПРЕДФРАКТАЛЬНОМ ГРАФЕ

Нижний Архыз
2004

УДК 519.1

АЛГОРИТМЫ С ОЦЕНКАМИ ПОСТРОЕНИЯ ПОКРЫТИЙ НЕПЕРЕСЕКАЮЩИМИСЯ ПРОСТЫМИ ЦЕПЯМИ НА ПРЕДФРАКТАЛЬНОМ ГРАФЕ

Д.А.Павлов, Р.А.Кочкаров

Карачаево-Черкесская государственная технологическая академия,
г. Черкесск, Карачаево-Черкесия, Россия, 369000

Аннотация. Статья посвящена многокритериальной задаче покрытия предфрактальных графов непересекающимися простыми цепями. Представлены алгоритмы выделяющие покрытия, оптимальное по определенным критериям и оцениваемые по остальным критериям. Все алгоритмы являются полиномиальными.
Ключевые слова: предфрактальный (n, L)-граф, подграф-затравка, паросочетание.

ALGORITHMS OF SOLUTION OF THE MULTICRITERIA PROBLEM OF A COVERING PREFRACTALS GRAPHS BY NON-INTERSECTED SIMPLE PATH

D.A.Pavlov, R.A.Kochkarov

Abstract. The paper is devoted to the multicriteria problem of covering prefractals graphs by non-intersecting simple chains. Algorithms selecting a covering, optimum by particular criteria and estimated according to remaining criteria, are represented. All the algorithms are polynomial.
Keywords: prefractal (n, L)-graph, subgraph - seeding agent, matching.

1. Введение. Основные определения

Исследуемая в настоящей работе задача покрытия предфрактальных графов простыми непересекающимися цепями содержательно относится к проблематике теории и методов автоматизации проектирования вычислительных систем. В настоящей работе задача покрытия предфрактальных графов простыми непересекающимися цепями рассматривается в многокритериальной постановке (Емеличев, Перепелица, 1989). Полученные оценки эффективности (Емеличев, Перепелица, 1994) также отражают многокритериальную суть поставленной задачи.

Оговоримся заранее, что недостающие определения и понятия теории графов и предфрактальных графов можно найти в работах (Кочкаров, 1998; Емеличев и др., 1990).

Введем понятие предфрактального графа. Пусть $H = (W, Q)$ – n-вершинный связный граф с множеством вершин W, $|W| = n$ и множеством ребер $|Q| = q$. Условимся называть его «затравкой». В качестве обобщения известной операции «расщепления вершины графа» (Емеличев и др., 1990) определим операцию «замещения вершины затравкой» (ЗВЗ). Для произвольного графа $G = (V, E)$ ее суть состоит в замещении вершины $v \in V$ графа n-вершинной затравкой $H = (W, Q)$. При этом каждое ребро $(v, v') \in E$, $v' \in V$, инцидентное вершине v, удаляется и заменяется на $(u, v') \in E$, где $u \in W$ – вершина затравки, выбираемая либо произвольно, либо по определенному закону (Кочкаров, 1998). Определим поэтапный процесс выполнения операции ЗВЗ. На этапе $l = 1$ в данной затравке $H = (W, Q)$ нумеруем вершины и ребра – получаем граф, обозначаемый через $G_1 = (V_1, E_1)$, т.е. $G_1 = (V_1, E_1) = H = (W, Q)$. Пусть выполнены этапы $l = 1,2,...,L$ и по завершении этапа L получен граф $G_L = (V_L, E_L)$ из $G_{L-1} = (V_{L-1}, E_{L-1})$ заменой каждой его вершины затравкой $H = (W, Q)$. Полученный граф будем называть предфрактальным (n, L)-графом, где $|W| = n$. Процесс порождения предфрактального графа G_L, по существу, есть процесс построения последовательности предфрактальных графов $G_1, G_2, ..., G_l, ..., G_L$, называемой *траекторией*.

Для предфрактального графа G_L ребра, появившиеся на l-ом, $l \in \{1,2,...,L\}$ этапе порождения, будем называть *ребрами ранга* l. *Новыми* ребрами предфрактального графа G_L назовем ребра ранга L, а все остальные ребра назовем *старыми*.

Если из предфрактального графа G_L, порожденного n-вершинной затравкой H, последовательно удалить все старые ребра (ребра ранга l, $l = 1,2,...,L-1$), то исходный граф

распадется на множество связных компонент $B_{l,s}^{(1)}$, $s=\overline{1,n^{l-1}}$, изоморфных (Емеличев и др., 1990) затравке H. Множество компонент $\{B_L^{(1)}\}$ будем называть *блоками первого ранга*. Аналогично, при удалении из предфрактального графа G_L всех старых ребер рангов $l=1,2,...,L-2$, получим множество *блоков* $\{B_L^{(2)}\}$ *второго ранга*. Обобщая скажем, что при удалении из предфрактального графа G_L всех ребер рангов $l=1,2,...,L-r$ получим множество $\{B_{L,i}^{(r)}\}$, $r\in\{1,2,...,L-1\}$, *блоком* r*-го ранга*, где $i=1,2,...,n^{L-r}$ – порядковый номер блока. Блоки $B_L^{(1)}\subseteq G_L$ первого ранга также будем называть *подграф-затравками* H предфрактального графа G_L.Очевидно, что всякий блок $B_L^{(r)}=(U_L^{(r)},M_L^{(r)})$, $r\in\{1,2,...,L-1\}$ является предфрактальным графом $B_r=(U_r,M_r)$, порожденным затравкой H.

Термином *подграф-затравка* $z_s^{(l)}$ будем называть блок $B_{l,s}^{(1)}$, $s=\overline{1,n^{l-1}}$ первого ранга предфрактального графа G_l, $l=\overline{1,L}$ из траектории. Последовательное выделение подграф-затравок $z_s^{(l)}$ на графах $G_1,G_2,...,G_L$ из траектории предфрактального графа G_L разбивает множество ребер E_L на непересекающиеся подмножества подграф-затравок $Z(G_L)=\{z_s^{(l)}\}$, где $l=\overline{1,L}$ – ранг подграф-затравки, а $s=\overline{1,n^{l-1}}$ – ее порядковый номер.

Будем говорить, что *предфрактальный граф* $G_L=(V_L,E_L)$ *взвешен*, если каждому его ребру $e^{(l)}\in E_L$ приписано действительное число $w(e^{(l)})\in(\theta^{l-1}a,\theta^{l-1}b)$, где $l=\overline{1,L}$ – ранг ребра, $a>0$, и $\theta<\frac{a}{b}$.

2. Многокритериальная постановка задачи о покрытии предфрактального графа простыми непересекающимися цепями

Рассмотрим взвешенный предфрактальный граф $G_L=(V_L,E_L)$, порожденный затравкой $H=(W,Q)$, у которой $|W|=n$, $|Q|=q$.

Под цепью предфрактального графа G_L понимаем простую цепь C (Емеличев и др., 1990), а количество вершин в ней называем *длиной цепи* и обозначаем $len(C)$. Простую цепь, ребра которой имеют одинаковый ранг l, будем называть l*-ранговой цепью* и обозначать $C^{(l)}$.

Покрытием вида $\aleph_1$ графа G_L будем называть подграф $y = (V_L, E_y)$, $E_y \subseteq E_L$, состоящий из множества простых цепей, где каждая вершина инцидентна ребрам только одной цепи из y. Это означает, что покрытие y состоит из компонент, каждая из которых является простой цепью. Всевозможные покрытия $\{y\}$ предфрактального графа G_L образуют множество допустимых решений $Y = Y(G_L) = \{y\}$ (МДР).

На множестве покрытий $y \in Y$ графа $G_L = (V_L, E_L)$ определены векторно-целевые функции (ВЦФ):

$$F(Y) = \{F(y) = (F_1(y), F_2(y), F_3(y)), y \in Y\} \quad (4)$$

$$F_1(y) = \sum_{C \in y} \sum_{e \in C} w(e) \to \min, \quad (5)$$

$\sum_{C \in y} \sum_{e \in C} w(e)$ – вес покрытия y;

$$F_2(y) = |y| \to \min, \quad (6)$$

где $|y|$ – мощность покрытия или количество цепей покрытия y;

$$F_3(y) = \hbar \to \min, \quad (7)$$

где – $\hbar$ количество типов цепей, составляющих покрытие y.

2.1. Алгоритм α_1 построения покрытия L-ранговыми цепями длины один

Алгоритм α_1 основан на алгоритме выделения совершенного паросочетания минимального веса, предложенный Эдмондсом (Кристофидес, 1978). Для целостного понимания опишем далее алгоритм Эдмондса и представим его в виде процедуры.

Алгоритм Эдмондса, выделения совершенного паросочетания минимального веса

На вход алгоритма Эдмондса подается произвольный взвешенный граф $G = (V, E)$, а на выходе получается совершенное паросочетание минимального веса. Для удобства формулировки алгоритма опишем задачу нахождения совершенного паросочетания максимального

веса (ЗМП). В свою очередь этот алгоритм будет основан на алгоритме для задачи нахождения наибольшего совершенного паросочетания (ЗНСП). Замена каждого веса на отрицательный позволит решать задачу нахождения совершенного паросочетания минимального веса. Приведенные ниже при формулировке алгоритма Эдмонса теоремы будут описаны без доказательства. Доказательство этих теорем можно найти в работе Кристофидеса (1978).

Представим далее некоторые определения, необходимые для описания алгоритма.

Если дано паросочетание M, то вершина v_j, не являющаяся конечной вершиной никакого ребра из M, будет называться экспонированной вершиной.

Альтернирующая относительно M цепь – это простая цепь, ребра которой попеременно лежат или не лежат в паросочетании M. Аугментальная относительно M цепь – это такая альтернирующая цепь, начальная и конечная вершины которой экспонированы.

Основная относящаяся к паросочетаниям теорема звучит следующим образом.

Теорема 1. *Паросочетание M будет наибольшим паросочетанием тогда и только тогда, когда в G не существует никакой аугменталъной относительно M цепи.*

Альтернирующим деревом относительно данного паросочетания *M* называется дерево T, для которого

(а) одна вершина, называемая корнем дерева T, является экспонированной;

(б) все начинающиеся в корне цепи являются альтернирующими;

(в) все максимальные цепи, начинающиеся в корне дерева T, содержат четное число ребер.

Разобьем все вершины дерева на два класса: Φ – класс внешних вершин, I – класс внутренних вершин. Корень дерева отнесем к классу Φ. Вершины вдоль любой цепи, начинающейся в корне, будут поочередно отнесены к классам Φ и I. Таким образом, если вершина расположена в “конце” цепи с нечетным числом ребер, начинающейся в корне, то эта вершина будет отнесена к I, а если она лежит в “конце” цепи с четным числом ребер, начинающейся в корне, то будет отнесена к Φ.

Аугменталъное дерево – это альтернирующее дерево относительно данного паросочетания, удовлетворяющее условию: существует ребро от какой-нибудь внешней вершины дерева v_0 до некоторой экспонированной вершины v_e, не принадлежащей дереву. Единственная цепь, идущая от корня дерева до вершины v_0 и далее – по ребру (v_0, v_e), будет аугментальной цепью.

Цветком по отношению к паросочетанию M называется аугментальная цепь, начальная и конечная экспонированные вершины которой совпадают, т.е. эта цепь является циклом, так как число ребер этой цепи нечетно.

В алгоритмах для ЗМП и ЗНПС, описанных ниже, цветки срезают для того, чтобы получить более простой новый граф. Срезание цветка B состоит в замене всех вершин цветка B (скажем, V_B) одной новой псевдовершиной v_b. Ребро (v_b, v_k) добавляется всегда, когда существует ребро из некоторой вершины $v_i \in V_B$ к другой вершине $v_k \notin V_B$. В упрощенном графе, полученном после срезания, вершина v_b и другие псевдовершины, соответствующие ранее срезанным цветкам, могут в свою очередь образовывать новый цветок, который опять срезается, и т.д. Последний цветок B_0, не содержащийся ни в каких других цветках, называется крайним цветком. Обоснование процесса срезания цветков основано на теореме 2, приводимой ниже.

Цветущее дерево – это альтернирующее дерево относительно данного паросочетания, для которого существует ребро (v_0', v_0''), соединяющее две внешние вершины дерева: v_0' и v_0''. Если P' – множество ребер цепи, начинающейся в корне альтернирующего дерева и кончающейся в v_0', а P'' – соответствующее множество для v_0'', то множество ребер $[P' \bigcup P'' - P' \bigcap P'']$ вместе с ребром (v_0', v_0'') образует цветок.

Когда цветок B срезан, получающаяся псевдовершина считается внешней вершиной. Всякая вершина из B может быть по желанию отнесена к внутренним или к внешним вершинам, так как если вершина принадлежит *B,* то она является концевой и в цепях с четным, и в цепях с нечетным числом ребер, начинающихся в корне дерева (в зависимости от маршрута, по которому проходят эти цепи до прихода в первую вершину цветка). Но когда псевдовершина, получающаяся после срезания цветка, помечается как “внешняя”, структура остающегося альтернирующего дерева все еще будет корректной – как это вытекает из определения альтернирующего дерева. Таким образом, после срезания цветка альтернирующее дерево в получившемся графе сохраняется.

Венгерское дерево – это такое альтернирующее дерево в графе, что каждое ребро графа, имеющее внешнюю вершину дерева в качестве концевой, имеет другой концевой вершиной внутреннюю вершину этого дерева. Важность венгерских деревьев для ЗМП объясняется следующим их свойством.

Теорема 2. *Пусть H – венгерское дерево в графе $G = (V, E)$ и $G_0 = (V - V_H)$ – порожденный подграф графа G, “исключающий” из G множество вершин V_H дерева H. Если M_H - паросочетание в дереве H и $M^*_{G_0}$ – наибольшее паросочетание в G_0, то множество ребер $M_H \cup M^*_{G_0}$ является наибольшим паросочетанием в G.*

Алгоритм для задачи нахождения наибольшего паросочетания (ЗНПС)

Пусть задано начальное паросочетание. (Допустимо использование пустого паросочетания.) Алгоритм осуществляет систематический поиск аугментальных цепей с целью улучшения заданного паросочетания в соответствии с теоремой 1.

Альтернирующее дерево имеет своим корнем некоторую экспонированную вершину и строится путем попеременного добавления ребер, лежащих и не лежащих в паросочетании, до тех пор, пока или

(I) дерево станет аугментальным, или

(II) на дереве появится цветок, или

(III) дерево станет венгерским.

В случае (I) число ребер в паросочетании может быть увеличено на единицу с помощью движения по аугментальной цепи к корню дерева и замены ребер цепи, принадлежащих паросочетанию, на ребра, не принадлежащие ему, и наоборот (при описанном построении получается новое паросочетание, состоящее из тех ребер выбранной аугментальной цепи, которые не принадлежали первоначальному паросочетанию). После этого дерево отбрасывается (дерево строится “отдельно от графа” и из графа не выбрасывается; если при построении дерева оно просто как-то выделялось (помечалось) в графе, то “выбрасывание” состоит в устранении всяких пометок, приписанных элементам дерева в этом процессе) и строится новое дерево, если такое существует, с корнем в некоторой оставшейся экспонированной вершине.

В случае (II) обнаруживается, что получившийся цветок срезается и продолжается рост дерева в процессе поиска аугментальной цепи. Что касается вычислительной стороны, то срезание не производится “явным образом”. Достаточно пометить все вершины цветка как внешние и отметить, что все они принадлежат этому цветку. Порядок, в котором цветки “срезаются”, важен, так как в конце всей процедуры цветки должны “расцвести” в обратном порядке.

В случае (III) вершины венгерского дерева и инцидентные им ребра совсем удаляются из графа, и алгоритм применяется к оставшемуся подграфу.

АЛГОРИТМ нахождения наибольшего паросочетания

Вход: взвешенный граф $G = (V, E)$.

Выход: наибольшее паросочетание $M = (V_M, E_M)$.

Шаг 1. Выбрать в G начальное паросочетание M.

Шаг 2. Если в G существуют, по крайней мере, две экспонированные вершины, то взять одну из них в качестве корня; в противном случае перейти к шагу 8.

Шаг 3. Взять внешнюю вершину v_0 дерева. Для каждого ребра (v_0, v_i): если v_i – экспонированная вершина, то перейти к шагу 4; если v_i – внутренняя вершина, то перейти к шагу 7; если v_i не принадлежит дереву и не является экспонированной, то перейти к шагу 5.

Шаг 4. Найдена аугментальная цепь из корня к v_i. Построить новое улучшенное паросочетание, удаляя текущее дерево и пометки вершин, и перейти к шагу 2.

Шаг 5. Добавить к альтернирующему дереву ребро (v_0, v_i) и отметить вершину v_i как внутреннюю. Найти ребро (v_i, v_k), принадлежащее текущему паросочетанию, и добавить его к дереву, пометив вершину v_k как внешнюю. Если существует ребро между v_k и другой внешней вершиной, то перейти к шагу 6. В противном случае перейти к шагу 3.

Шаг 6. Опознать и срезать получившийся цветок. Отметить возникшую псевдовершину как внешнюю. Внести поправку в частичное упорядочение цветков и возвратиться к шагу 3.

Шаг 7. Возвращаться к шагу 3 до тех пор, пока единственным возникающим случаем не будет случай (III). Тогда удалить все вершины дерева и все ребра из G, инцидентные этим вершинам. Считать оставшийся подграф графом G и вернуться к шагу 2.

Шаг 8. Выделить отдельно в последнем графе G и в каждом удаленном венгерском графе оптимальное паросочетание, поступая так. Распустить сначала крайний цветок и выделить паросочетание, которое оставляет экспонированной относительно него ту вершину v, которая входит в паросочетание в нераспустившемся цветке. Продолжать процесс распускания в порядке, обратном к установленному на шаге 6, до тех пор, пока весь граф не будет распустившимся и не будет получено наибольшее паросочетание.

Алгоритм для задачи нахождения максимального паросочетания (совершенного паросочетания максимального веса)

Предположим, что мы начинаем с графа G. Присвоим его вершинам веса π_i, выбирая величины π_i достаточно большими, когда все λ_r взяты равными нулю. Пусть теперь G' – остовный подграф в G. Назовем G' специальным остовным подграфом графа *G.* Если в *G′* можно найти совершенное паросочетание (с помощью алгоритма для ЗНПС, описанного выше), то это паросочетание – оптимальное.

Вообще говоря, может оказаться, что в G' совершенное паросочетание найти нельзя, а алгоритм для ЗНПС обнаружит венгерское дерево. Как уже отмечалось, наибольшее паросо-

четание в G' частично строится из паросочетания в H, оставляющего экспонированным корень дерева H, и поэтому существование венгерского дерева означает, что в G' нет никакого совершенного паросочетания. Таким образом, при обнаружении такого дерева следует изменить вектор весов $[\pi_i, \lambda_r]$, чтобы получить новый специальный остовный подграф G' в графе G и подвергнуть его проверке. Ниже будет описано, как вычисляется такой весовой вектор.

Для более ясного понимания алгоритма мы введем следующую индексацию. Первоначальный граф G будет обозначаться через G_0. Специальный остовный подграф графа G_0 есть G'_1 ; в этом графе строится альтернирующее дерево, как это делалось в алгоритме для ЗНПС. Когда, наконец, образуется и срезается первый цветок, сам граф будет обозначаться через G_1, а его специальный остовный подграф – через G'_2 . После срезания f цветков граф будет обозначен символом G_f, а его специальный остовный подграф – G'_{f+1}.

Общий шаг алгоритма таков. Пусть очередной граф (после срезания некоторого числа цветков) будет G_{f-1}, а его специальный остовный подграф – G'_f. В G'_f строится альтернирующее дерево – с помощью алгоритма ЗНПС – до тех пор, пока не произойдет одно из трех:

А) на дереве появится цветок;

Б) дерево станет аугментальным;

В) дерево станет венгерским.

Случай А. В этом случае цветок срезается, получается новый граф G'_f и его специальный остовный подграф G'_{f+1}. Как отмечалось выше, срезание цветка приводит к дереву T с корректной структурой альтернирующего дерева, так что T можно сохранить и продолжать процесс роста дерева.

Случай Б. В этом случае, как уже объяснялось для ЗНПС, получается улучшенное паросочетание – с большим числом ребер.

Дерево T следует отбросить и, взяв в графе G'_f оставшуюся экспонированную (в G'_f) вершину, приступить к построению нового дерева с корнем в этой вершине. Здесь следует заметить, что как только T отброшено и в G'_f “начинает расти новое дерево”, некоторые из псевдовершин в G'_f (образованные после срезания более ранних цветков, давших графы $G_0, G_1, ..., G_{f-1}$) могут оказаться помеченными как внутренние в новом дереве.

Случай В. В этом случае вектор весов $[\pi_i, \lambda_r]$ для текущего графа G_{f-1} изменяется так, что возникает новый специальный остовный подграф, отличный от G'_f. Изменения в $[\pi_i, \lambda_r]$ делаются так, чтобы

(а) новый граф G'_f продолжал удовлетворять сделанным ранее предположениям, т. е. что λ_r отлично от нуля лишь для множеств вершин, соответствующих псевдовершинам из G_{f-1} и что равенство выполняется для всех ребер или цветков, которые после срезания дали псевдовершины в G_{f-1}.

(б) текущее альтернирующее дерево в старом графе G'_f можно было строить дальше, или на нем возникнет цветок, или оно станет аугментальным – с использованием новых ребер, входящих в новый граф G'_f; либо некоторая псевдовершина в текущем альтернирующем дереве, помеченная как внутренняя, перестанет быть таковой; либо будет доказано, что в G_0 нет совершенного паросочетания.

Алгоритм завершает работу, когда на некотором этапе f в графе G'_f будет получено совершенное паросочетание. Соответствующий граф G_{f-1} расширяется до начального графа G_0 путем распускания цветков в порядке, обратном порядку их срезания, и построения паросочетаний в каждом распустившемся цветке – так же, как на шаге 8 алгоритма ЗНПС.

Когда получено совершенное паросочетание, оно будет максимальным совершенным паросочетанием.

АЛГОРИТМ нахождения максимального паросочетания

Вход: взвешенный граф $G = (V, E)$.

Выход: Максимальное паросочетание $M = (V_M, E_M)$.

Шаг 1. Присвоить вершинам графа G_0 веса π_i так, чтобы для всех ребер из G_0 было выполнено соотношение $\pi_i + \pi_k + \sum_{S_r \in F_j} \lambda_r \geq c_j$, $\forall j = \overline{1, m}$, когда все $\lambda_r = 0$. Взять $G = G_0$.

Шаг 2. Образовать специальный остовный подграф G' текущего графа G.

Шаг 3. Если в G' не существует никакого альтернирующего дерева T, то строить новое дерево с корнем в некоторой экспонированной вершине из G'. Если экспонированных вершин нет, перейти к шагу 8. Если в G' уже существует альтернирующее дерево, то продолжать

его построение. Если на T обнаружится цветок, то перейти к шагу 4. Если дерево T станет аугментальным, то перейти к шагу 5.

Если дерево T станет венгерским, то перейти к шагу 6.

Шаг 4. Срезать цветки и образовать псевдовершину. Обозначим получившийся граф через G, его специальный остовный подграф через G', а оставшееся дерево через T. Перейти к шагу 3.

Шаг 5. Улучшить текущее паросочетание, "поменяв" вдоль аугментальной цепи ребра, принадлежащие паросочетанию, и не принадлежащие ему. "Отбросить" дерево T и перейти к шагу 3.

Шаг 6. Вычислить Δ_1, Δ_2, Δ_3 и Δ по формулам

$$\Delta_1 = \min_{e_j}[\pi_i + \pi_k - c_j],$$

$$\Delta_2 = \frac{1}{2}\min_{e_j}[\pi_i + \pi_k - c_j],$$

$$\Delta_3 = \frac{1}{2}\min_{S_r}[\lambda_r],$$

$$\Delta = \min[\Delta_1, \Delta_2, \Delta_3].$$

Если нет никаких ограничений Δ_1, Δ_2, Δ_3, то остановиться; в графе нет совершенного паросочетания. В противном случае изменить вектор весов $[\pi_i, \lambda_r]$, как описано выше. Если $\Delta = \Delta_1$ или $\Delta = \Delta_2$, то сохранить текущее дерево T и перейти к шагу 2. Если $\Delta = \Delta_3$, то перейти к шагу 7.

Шаг 7. "Распустить" псевдовершину, давшую цветок B в соотношении Δ_3. Обозначим получившийся граф через G, а его специальный остовный подграф через G'. Построить в B совершенное паросочетание. Перестроить альтернирующее дерево, добавляя к ребрам из T необходимую цепь из B, и обозначить это дерево символом T. Перейти к шагу 3.

Шаг 8. Распустить все цветки в обратном порядке (к первоначальному порядку срезания цветков), находя совершенное паросочетание после каждого распускания.

Изменив веса графа $G = (V, E)$ в алгоритме нахождения максимального паросочетания на отрицательные, получим алгоритм нахождения минимального паросочетания (совершенного паросочетания минимального веса). Далее алгоритм Эдмондса будем использовать как процедуру Edmonds, сохраняя все внутренние шаги алгоритма.

Алгоритм α_1

Рассмотрим взвешенный предфрактальный граф $G_L = (V_L, E_L)$, порожденный затравкой $H = (W, Q)$, у которой $|W| = n$, $|Q| = q$.

Алгоритм α_1 строит покрытие $y_1 = (V, E_{y_1})$ цепями длины ребра, оптимальное по критерию $F_1(y_1) = \sum_{C \in y_1} \sum_{e \in C} w(e) \to \min$.

Теорема 3. *Для того чтобы предфрактальный (n, L)-граф $G = (V, E)$, порожденный затравкой $H = (W, Q)$, являлся факторизуемым, необходимо и достаточно существование совершенного паросочетания $H_M = (W, Q_M)$ на самой затравке $H = (W, Q)$.*

Данная теорема гласит о том, что если удастся найти совершенное паросочетание на порождающей затравке $H = (W, Q)$, тогда обязательно найдется совершенное паросочетание $M = (V, E_M)$ и на самом предфрактальном графе $G = (V, E)$. Таким образом, предположим, что выполняется условие теоремы 2.3., и алгоритм α_1 заведомо выделит совершенное паросочетание минимального веса.

Опишем принцип работы алгоритма α_1.

Основная идея алгоритма заключается в том, что каждая подграф-затравка $z_s^{(L)}$, $s = \overline{1, n^{L-1}}$ рассматривается как отдельно взятый граф. Последовательно на каждой из n^{L-1} подграф-затравке находятся ее совершенные паросочетания M_s. Поиск совершенного паросочетания на отдельно взятой подграф-затравке осуществляется с помощью алгоритма Эдмондса, описанного ранее. Алгоритм Эдмондса применяется в алгоритме α_1 как процедура. Осуществив поиск совершенных паросочетаний минимального веса на подграф-затравках, мы получим совершенное паросочетание минимального веса всего предфрактального графа G_L, после чего алгоритм α_1 заканчивает свою работу.

АЛГОРИТМ α_1

Вход: взвешенный предфрактальный граф $G_L = (V_L, E_L)$.

Выход: совершенное паросочетание минимального веса M_L.

Шаг 1. Последовательно для каждой затравки $z_s^{(L)}$, $s = \overline{1, n^{L-1}}$ найти совершенное паросочетание M_s, используя процедуру Эдмондса.

Шаг 2. На выходе шага 1 получаем $s = \overline{1, n^{L-1}}$ совершенных паросочетаний минимального веса M_s для каждой затравки $z_s^{(L)}$. Объединяя эти совершенные паросочетания M_s, получим совершенное паросочетание минимального веса M_L предфрактального графа G_L.

Процедура Эдмондса.

Вход: взвешенный граф $G = (V, E)$.

Выход: СПМВ $M = (V, E_M)$.

Теорема 4. *Алгоритм α_1 строит покрытие цепями длины один (ребро) на предфрактальном (n, L)-графе $G_L = (V_L, E_L)$ оптимальное по критерию $F_1(x)$, если $\theta < \frac{a}{b}$.*

Доказательство. Поиск совершенных паросочетаний минимального веса на подграф-затравках осуществляется с помощью алгоритма Эдмондса.

На последнем шаге L построения предфрактального графа G_L все вершины замещаются затравкой $H = (W, Q)$, тогда каждая вершина *L*-ого ранга принадлежит какой-либо подграф-затравке $z_s^{(L)}$. Найдя совершенное паросочетание на затравке $z_s^{(L)}$, покроются все вершины, принадлежащие этой затравке.

Таким образом отбросим все ребра ранга $l = \overline{1, L-1}$, и будем работать только с подграф-затравками *L*-ого ранга $z_s^{(L)}$, $s = \overline{1, n^{L-1}}$.

Выделим на подграф-затравке *L*-ого ранга $z_1^{(L)}$ совершенное паросочетание минимального веса M_1. В результате паросочетание M_1 покрыло *n* вершин. Далее выделим на второй затравке *L*-ого ранга $z_2^{(L)}$ совершенное паросочетание минимального веса M_2. Паросочетание M_2 покрыло еще *n* вершин. Это покрытие никак не повлияло на M_1, так как затравки $z_1^{(L)}$ и $z_2^{(L)}$ являются отдельными, не связанными между собой подграфами.

Тогда, после выделения совершенного паросочетания $M_{n^{L-1}}$ на последней затравке $z_{n^{L-1}}^{(L)}$, будет покрыто $n^{L-1} \cdot n = n^L$ вершин. Получим, что покрыто все множество вершин $G_L = (V_L, E_L)$: $|V_L| = n^L$.

На затравках $z_s^{(L)}$ выделялись совершенные паросочетания M_s, $s = \overline{1, n^{L-1}}$, то есть вершины предфрактального графа покрывались паросочетанием. Таким образом, полученное покрытие является совершенным паросочетанием M_L предфрактального графа G_L. А по-

скольку предфрактальный граф взвешен с помощью коэффициента $\theta < \frac{a}{b}$, вес ребра L-го ранга меньше веса любого из ребер предыдущих рангов. Таким образом, поиск совершенных паросочетаний минимального веса производился на ребрах самых минимальных весов, следовательно, совершенное паросочетание M_L, выделенное на предфрактальном графе G_L, будет минимального веса.

ТЕОРЕМА 5. *Вычислительная сложность алгоритма α_1, покрытия цепями длины один (ребро) минимального веса на предфрактальном (n,L)-графе $G_L = (V_L, E_L)$, порожденного затравкой $H = (W, Q)$, где $|W| = n$, $|V_L| = N = n^L$, равна $O(N \cdot n^2)$.*

ДОКАЗАТЕЛЬСТВО. Алгоритм α_1 представляет собой, по существу многократное выполнение шага 1, т.е. поиск паросочетания для каждой подграф-затравки, а их n^{L-1}. Шаг 1 потребует выполнения $O(n^3)$ операций на каждой подграф-затравке – столько операций выполняет алгоритм Эдмондса.

Тогда $O(n^{L-1} \cdot n^3) = O(n^L \cdot n^2) = O(N \cdot n^2)$. Таким образом, вычислительная сложность алгоритма α_1 равна $O(N \cdot n^2)$.

Сравнив вычислительную сложность алгоритма α_1 с вычислительной сложностью алгоритма Эдмондса, получим: $O(N \cdot n^2) < O(N^3)$.

ПРИМЕЧАНИЕ 1. *Вычислительная сложность алгоритма α_1 меньше вычислительной сложности алгоритма Эдмондса в N^2 раз на предфрактальном (n,L)-графе $G_L = (V_L, E_L)$.*

ТЕОРЕМА 6. *Алгоритм α_1 выделяет покрытие $y_1 = (V, E_{y_1})$ на предфрактальном (n,L)-графе $G_L = (V_L, E_L)$, порожденном n-вершинной затравкой $H = (W, Q)$, оптимальное по первому $F_1(y_1)$ и третьему $F_3(y_1)$ критерию, и оцениваемое по второму, $F_2(y_1) \le \frac{n^L}{2}$.*

ДОКАЗАТЕЛЬСТВО. Доказательством оптимальности по первому критерию $F_1(y_1)$ служит теорема 2.4, поскольку на выходе алгоритма α_1 получаем совершенное паросочетание минимального веса. Третий критерий $F_3(y_1)$ минимизирует количество типов цепей, т.е. число цепей различных длин, но покрытие состоит только из цепей длины ребра, тогда критерий

$F_3(y_1)$ принимает заведомо минимальное число, $F_3(y_1)=1$. Критерий $F_2(y)\le|y|$, мощность покрытия, оценивается следующим образом. На одной затравке мощность покрытия не больше $\frac{n}{2}$, так как количество вершин на затравке равно n. А поскольку алгоритм α_1 работает с подграф-затравками L-го ранга, а их n^{L-1}, то получаем $F_2(y)\le\frac{n}{2}\cdot n^{L-1}=\frac{n^L}{2}$.

2.2. Алгоритм α_2 построения покрытия $\aleph_1$ L-ранговыми цепями длины два (два ребра)

Построим алгоритм α_2 покрытия цепями длины два (два ребра) на взвешенном предфрактальном (n,L)-графе $G_L=(V_L,E_L)$ порожденном затравкой $H=(W,Q)$, у которой $|W|=n$, $|Q|=q$, где n кратно трем, т.е. алгоритм α_2 строит покрытие $y_2=(V,E_{y_2})$ простыми цепями длины два ребра.

Опишем вначале словесно принцип работы алгоритма α_2.

Основная идея алгоритма заключается в том, что каждая подграф-затравка $z_s^{(L)}$, $s=\overline{1,n^{L-1}}$ рассматривается как отдельно взятый граф. Последовательно на каждой из n^{L-1} подграф-затравке производится поиск покрытий простыми цепями длины два M_s. Поиск покрытия на отдельно взятой подграф-затравке осуществляется с помощью процедуры Выделения покрытия цепями длины два, в основе которой лежит алгоритм Эдмондса, описанный ранее. Осуществив поиск покрытий простыми цепями длины два на подграф-затравках, мы получим покрытие $y_2=(V,E_{y_2})$ простыми цепями длины два ребра предфрактального графа G_L, после чего алгоритм α_2 заканчивает свою работу.

Предложим краткое описание процедуры Выделения покрытия цепями длины два для целостного понимания алгоритма α_2. На взвешенном графе $G=(V,E)$ производится поиск совершенного паросочетания минимального веса M_1, используя описанный ранее алгоритм Эдмондса. Далее среди элементов совершенного паросочетания минимального веса M_1, выделяется элемент $m_j\in M_1$, $j\in J$ с максимальным весом. Далее, рассматривая у концевых вершин выбранного элемента m_j инцидентные ребра графа G, выбираем и выделяем те из них, которые имеют минимальный вес (у каждой концевой вершины покрытия их может быть несколько). Выделенные ребра уже соединяют два элемента покрытия $m_i,m_j\in M_1$,

$i \in I$ и образует в отдельности друг от друга цепи c_3. Далее рассматриваем в отдельности каждую из этих цепей и выбираем среди них ту, которая имеет минимальный вес. У выбранной минимальной цепи c_3^* сравниваем и выбираем из двух ребер принадлежащих покрытию M_1 то, которое имеет минимальный вес, а ребро покрытия M_1 с максимальным весом исключаем из цепи c_3^*, запоминая и окрашивая конечную вершину исключенного ребра, принадлежащего M_1. В результате этой операции мы выделили цепь длины два. Далее описанная операция применяется многократно, до тех пор пока не будет покрыт весь граф G цепями длины два.

АЛГОРИТМ α_2

ВХОД: взвешенный предфрактальный граф $G_L = (V_L, E_L)$.

ВЫХОД: покрытие $y_2 = (V, E_{y_2})$ простыми цепями длины два.

ШАГ 1. Последовательно для каждой затравки $z_s^{(L)}$, $s = \overline{1, n^{L-1}}$ найти покрытие M_s простыми цепями длины два, используя процедуру ПЦДД.

ШАГ 2. На выходе шага 1 получаем $s = \overline{1, n^{L-1}}$ покрытий M_s, для каждой затравки $z_s^{(L)}$. Объединяя эти покрытия M_s получим покрытие $y_2 = (V_L, E_{y_2})$ предфрактального графа G_L.

ПРОЦЕДУРА ВЫДЕЛЕНИЯ ПОКРЫТИЯ ЦЕПЯМИ ДЛИНЫ ДВА (ПЦДД).

ВХОД: взвешенный граф $G = (V, E)$.

ВЫХОД: покрытие M простыми цепями длины два.

ТЕОРЕМА 7. *Если существует покрытие, то алгоритм* α_2 *выделяет покрытие* $y_2 = (V, E_{y_2})$ цепями длины два (два ребра) *на предфрактальном* (n, L)*-графе* $G_L = (V_L, E_L)$, порожденном n*-вершинной затравкой* $H = (W, Q)$*, где* n *кратно трем.*

ДОКАЗАТЕЛЬСТВО. Поиск цепей длины два на подграф-затравках осуществляется с помощью процедуры выделения покрытия цепями длины два, основанной на алгоритме Эдмондса.

На последнем шаге L построения предфрактального графа G_L все вершины замещаются затравкой $H = (W, Q)$, тогда каждая вершина *L*-ого ранга принадлежит какой-либо подграф-затравке $z_s^{(L)}$. Таким образом, в результате выделения покрытия на подграф-затравке будут покрыты все вершины, принадлежащие этой затравке.

Поэтому алгоритм α_2 работает только с подграф-затравками L-ого ранга $z_s^{(L)}$, $s = \overline{1, n^{L-1}}$.

Выделим на подграф-затравке L-ого ранга $z_1^{(L)}$ покрытие M_1. В результате покрытие M_1 выделило n вершин. Далее выделим на второй затравке L-ого ранга $z_2^{(L)}$ покрытие M_2. Покрытие M_2 выделило еще n вершин. Это покрытие никак не повлияло на M_1, так как подграф-затравки $z_1^{(L)}$ и $z_2^{(L)}$ являются отдельными, не связанными между собой подграфами.

Тогда, после выделения покрытия $M_{n^{L-1}}$ на последней подграф-затравке $z_{n^{L-1}}^{(L)}$, будет покрыто $n^{L-1} \cdot n = n^L$ вершин. Получим, что покрыто все множество вершин $G_L = (V_L, E_L) : |V_L| = n^L$.

На затравках $z_s^{(L)}$ выделялись покрытия M_s, $s = \overline{1, n^{L-1}}$, то есть вершины предфрактального графа покрывались цепями длины два. Таким образом, полученное покрытие является покрытием $y_2 = (V, E_{y_2})$ предфрактального графа G_L.

Теорема 8. *Вычислительная сложность алгоритма α_2 покрытия цепями длины два (два ребра) на предфрактальном (n, L)-графе $G_L = (V_L, E_L)$, порожденном затравкой $H = (W, Q)$, где $|W| = n$, $|V_L| = N = n^L$, равна $O(N \cdot 2n^2)$.*

Доказательство. Алгоритм α_2 представляет собой, по существу многократное выполнение шага 1, т.е. поиск покрытия длины два ребра для каждой подграф-затравки, а их n^{L-1}. Для каждой затравки выполняются следующие действия. Сначала производится поиск совершенного паросочетания минимального веса, что требует выполнения $O(n^3)$ операций на каждой подграф-затравке, столько операций выполняет алгоритм Эдмондса. Далее просматриваются все ребра паросочетания и находится наибольшее, т.е. выполняется $\frac{n}{2}$ операций. Далее просматриваются все смежные ребра для каждого ребра паросочетания, если затравка полный граф для каждой вершины будет просмотрено $(n-1)$ ребер, в сумме будет выпол-

нено $\frac{n}{2} \cdot 2(n-1) = n(n-1)$ операций. Затем для каждого элемента покрытия цепями два ребра будет просмотрено по две цепи длины три ребра, что составит $\frac{2n}{3}$ операций.

Таким образом, получим $\frac{n}{2} + n(n-1) + \frac{2n}{3} = n^2 + \frac{n}{2} + \frac{2n}{3} - n = n^2 + \frac{n}{6} \le n^3$. Тогда количество операций на затравке будет равняться $n^3 + n^3 = 2n^3$.

Тогда вычислительная сложность алгоритма α_2 на всем предфрактальном графе равна $O(n^{L-1} \cdot 2n^3) = O(n^L \cdot 2n^2) = O(N \cdot 2n^2)$. Таким образом, вычислительная сложность алгоритма α_2 равна $O(N \cdot 2n^2)$.

Теорема 9. *Алгоритм α_2 выделяет покрытие $y_2 = (V, E_{y_2})$ на предфрактальном (n, L)-графе $G_L = (V_L, E_L)$, порожденном n-вершинной затравкой $H = (W, Q)$, оптимальное по третьему $F_3(y_2)$ критерию и оцениваемое по первому критерию, $F_1(y_2) \le \frac{2n^L}{3}\theta^{L-1}b$ и по второму $F_2(y_2) \le \frac{n^L}{3}$, если $\theta < \frac{a}{b}$.*

Доказательство. Третий критерий $F_3(y_2)$ минимизирует количество типов цепей, т.е. число цепей различных длин, но покрытие состоит только из цепей длины два ребра, тогда критерий $F_3(y_2)$ принимает заведомо минимальное число, $F_3(y_2) = 1$. Критерий $F_2(y_2) = |y_2|$, мощность покрытия, оценивается следующим образом. На одной затравке мощность покрытия не превышает $\frac{n}{3}$, так как количество вершин на затравке равно n. А поскольку алгоритм α_2 работает с подграф-затравками L-го ранга, а их n^{L-1}, то получаем $F_2(y) \le \frac{n}{3} \cdot n^{L-1} = \frac{n^L}{3}$. Первый критерий оценивается следующим образом. Так как предфрактальный граф G_L взвешен с помощью коэффициента подобия $\theta < \frac{a}{b}$, тогда веса ребер L-го ранга $w(e^{(L)})$ будут принадлежать интервалу $(\theta^{L-1}a, \theta^{L-1}b)$. Предположим, что выделенное покрытие включает в себя ребра самых больших весов $\theta^{L-1}b$, тогда вес покрытия

будет равен количеству элементов покрытия, помноженному на их вес, т.е. $F_1(y_2) \le \frac{n^L}{3} \cdot 2(\theta^{L-1}b) = \frac{2n^L}{3}\theta^{L-1}b$.

2.3. Алгоритм α_3 построения покрытия $\aleph_1$ L-ранговыми цепями длины три (три ребра)

Рассмотрим также взвешенный предфрактальный граф $G_L = (V_L, E_L)$, порожденный затравкой $H = (W, Q)$, у которой $|W| = n$, $|Q| = q$, где n кратно четырем. Алгоритм α_3 строит покрытие $y_3 = (V, E_{y_3})$ простыми цепями длины три (три ребра).

Опишем словесно принцип работы алгоритма α_3. Как и для алгоритма α_2 каждую подграф-затравку $z_s^{(L)}$, $s = \overline{1, n^{L-1}}$ будем рассматривать как отдельно взятый граф. Последовательно на каждой из n^{L-1} подграф-затравок производится поиск покрытий простыми цепями длины три ребра. Поиск покрытия на отдельно взятой подграф-затравке осуществляется с помощью процедуры Выделения покрытия цепями длины три, в основе которой лежит алгоритм Эдмондса. Осуществив поиск покрытий простыми цепями длины три на подграф-затравках, мы получим покрытие $y_3 = (V, E_{y_3})$ простыми цепями длины три ребра предфрактального графа G_L, после чего алгоритм α_3 заканчивает свою работу.

Предложим краткое описание процедуры Выделения покрытия цепями длины три для целостного понимания алгоритма. На взвешенном графе $G = (V, E)$ производится поиск совершенного паросочетания минимального веса M_1. Запоминаем далее первоначальную структуру графа. Затем каждый элемент покрытия $m_j \in M_1$ стягиваем в вершину.

К полученному в результате операции стягивания покрытых ребер новом графе применяется снова алгоритм α_1 покрытия предфрактального графа цепями длины один. В результате чего имеем покрытие M_2, которое также выделяем. Далее восстанавливается первоначальная структура графа $G = (V, E)$, на котором уже будут выделены покрытия M_1 и M_2. Объединение множеств ребер, входящих в M_1 и M_2 ($M_1 \cup M_2$), будут образовывать цепь длины три.

АЛГОРИТМ α_3

Вход: взвешенный предфрактальный граф $G_L = (V_L, E_L)$.

ВЫХОД: покрытие $y_3 = (V, E_{y_3})$ простыми цепями длины три.

ШАГ 1. Последовательно для каждой затравки $z_s^{(L)}$, $s = \overline{1, n^{L-1}}$ найти покрытие M_s простыми цепями длины три, используя процедуру ПЦДТ.

ШАГ 2. На выходе шага 1 получаем $s = \overline{1, n^{L-1}}$ покрытий M_s для каждой затравки $z_s^{(L)}$. Объединяя эти покрытия M_s, получим покрытие $y_3 = (V, E_{y_3})$ предфрактального графа G_L.

ПРОЦЕДУРА ВЫДЕЛЕНИЯ ПОКРЫТИЯ ЦЕПЯМИ ДЛИНЫ ТРИ РЕБРА (ПЦДТ)

ВХОД: взвешенный граф $G = (V, E)$.

ВЫХОД: покрытие M простыми цепями длины три ребра.

ТЕОРЕМА 10. *Если существует покрытие, то алгоритм* α_3 *на предфрактальном* (n, L)*-графе* $G_L = (V_L, E_L)$ *выделяет покрытие* $y_3 = (V, E_{y_3})$.

ДОКАЗАТЕЛЬСТВО. Поиск цепей длины три ребра на подграф-затравках осуществляется с помощью процедуры выделения покрытия цепями длины три, основанной на алгоритме Эдмондса.

На последнем шаге L построения предфрактального графа G_L все вершины замещаются затравкой $H = (W, Q)$, тогда каждая вершина *L*-ого ранга принадлежит какой-либо подграф-затравке $z_s^{(L)}$. Таким образом, в результате выделения покрытия на подграф-затравке будут покрыты все вершины, принадлежащие этой затравке.

Поэтому алгоритм α_3 работает только с подграф-затравками *L*-ого ранга $z_s^{(L)}$, $s = \overline{1, n^{L-1}}$.

Выделим на подграф-затравке *L*-ого ранга $z_1^{(L)}$ покрытие M_1. В результате покрытие M_1 выделило *n* вершин. Далее выделим на второй затравке *L*-ого ранга $z_2^{(L)}$ покрытие M_2. Покрытие M_2 выделило еще *n* вершин. Это покрытие никак не повлияло на M_1, так как подграф-затравки $z_1^{(L)}$ и $z_2^{(L)}$ являются отдельными, не связанными между собой подграфами.

Тогда, после выделения покрытия $M_{n^{L-1}}$ на последней подграф-затравке $z_{n^{L-1}}^{(L)}$, будет покрыто $n^{L-1} \cdot n = n^L$ вершин. Получим, что покрыто все множество вершин $G_L = (V_L, E_L)$: $|V_L| = n^L$.

На затравках $z_s^{(L)}$ выделялись покрытия M_s, $s = \overline{1, n^{L-1}}$, то есть вершины предфрактального графа покрывались цепями длины три. Таким образом, полученное покрытие является покрытием $y_3 = (V_L, E_{y_3})$ предфрактального графа G_L.

Теорема 11. *Вычислительная сложность алгоритма α_3 выделения покрытия цепями длины три (три ребра) на предфрактальном (n, L)-графе $G_L = (V_L, E_L)$, порожденном затравкой $H = (W, Q)$, где $|W| = n$, $|V_L| = N = n^L$, равна $O(N \cdot 2n^2)$.*

Доказательство. Алгоритм α_3 представляет собой, по существу многократное выполнение шага 1, т.е. поиск покрытия длины три ребра для каждой подграф-затравки, а их n^{L-1}. Для каждой затравки выполняются следующие действия. Сначала производится поиск совершенного паросочетания минимального веса, что требует выполнения $O(n^3)$ операций на каждой подграф-затравке, столько операций выполняет алгоритм Эдмондса. Далее, после стягивания ребер паросочетания, снова производится поиск совершенного паросочетания минимального веса.

Таким образом, получим $n^3 + n^3 = 2n^3$ операций на затравке. Тогда вычислительная сложность алгоритма α_3 на всем предфрактальном графе равна $O(n^{L-1} \cdot 2n^3) = O(n^L \cdot 2n^2) = O(N \cdot 2n^2)$. Таким образом, вычислительная сложность алгоритма α_3 равна $O(N \cdot 2n^2)$.

Теорема 12. *Алгоритм α_3 выделяет покрытие $y_3 = (V, E_{y_3})$* цепями длины три (три ребра) *на предфрактальном (n, L)-графе $G_L = (V_L, E_L)$, порожденном n-вершинной затравкой $H = (W, Q)$, оптимальное по третьему $F_3(y_3)$ критерию и оцениваемое по первому критерию $F_1(y_3) \le \frac{3n^L}{4}\theta^{L-1}b$ и по второму $F_2(y_3) = \frac{n^L}{4}$.*

Доказательство. Третий критерий $F_3(y_3)$ минимизирует количество типов цепей, т.е. число цепей различных длин, но покрытие состоит только из цепей длины два ребра, тогда критерий $F_3(y_3)$ принимает заведомо минимальное число, $F_3(y_3) = 1$. Критерий $F_2(y_3) = |y_3|$, мощность покрытия, оценивается следующим образом. На одной затравке мощность покрытия равна $\frac{n}{4}$, так как количество вершин на затравке равно n. А поскольку алгоритм α_3 работает с подграф-затравками L-го ранга, а их n^{L-1}, то получаем

$F_2(y_3) = \frac{n}{4} \cdot n^{L-1} = \frac{n^L}{4}$. Первый критерий оценивается следующим образом. Так как предфрактальный граф G_L взвешен с помощью коэффициента подобия $\theta < \frac{a}{b}$, тогда веса ребер L-го ранга $w(e^{(L)})$ будут принадлежать интервалу $(\theta^{L-1}a, \theta^{L-1}b)$. Предположим, что выделенное покрытие включает в себя ребра самых больших весов $\theta^{L-1}b$, тогда вес покрытия будет равен количеству элементов покрытия, помноженному на их вес, т.е. $F_1(y_2) \le \frac{n^L}{4} \cdot 3(\theta^{L-1}b) = \frac{3n^L}{4}\theta^{L-1}b$.

Литература

Емеличев В.А., Перепелица В.А. О некоторых алгоритмических проблемах многокритериальной оптимизации на графах . Журнал вычисл. матем. и матем. физ. 1989, № 2. С. 171 – 183.

Емеличев В.А., Перепелица В.А. Сложность дискретных многокритериальных задач .Дискретная математика. – 1994. Т. 6, вып. 1. С .3 – 33.

Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. – М.: Наука, 1990.

Кочкаров А.М. Распознавание фрактальных графов. Алгоритмический подход. – Нижний Архыз, ПрепринтСАО №133, 1998.

Кристофидес Н. Теория графов. Алгоритмический подход. - М., Мир, 1978.

Павлов Д.А. Многокритериальная задача покрытия предфрактального графа цепями типа τ ($\tau = 1,2,3$). Черкесск 2003г. Деп. в ВИНИТИ №670-В2003. С.1-17.

Бесплатно

Д.А.Павлов, А.А.Кочкаров

АЛГОРИТМЫ С ОЦЕНКАМИ ПОСТРОЕНИЯ ПОКРЫТИЙ НЕПЕРЕСЕКАЮЩИМИСЯ ПРОСТЫМИ ЦЕПЯМИ НА ПРЕДФРАКТАЛЬНОМ ГРАФЕ

Работа поступила в печать 9 сентября 2004 г.

Заказ № 157 с Уч.изд.л.-2.9 Тираж 25
Специальная астрофизическая обсерватория РАН