В статье проведен критический анализ метода расчета параметра эффективности маскирования речи в технических каналах утечки информации. Предложен и подтвержден экспериментальными исследованиями метод цифровой корреляционной обработки контрольного фрагмента речи, который устраняет недостатки существующего метода и позволяет на основе расчета коэффициента корреляции АКФ сегментов сигнала контрольного слова и сигнала канала утечки рассчитать параметр эффективности аддитивного маскирования речевых сигналов.

ПЕРЕЧЕНЬ ССЫЛОК

1. *Дворянкин С. В., Макаров Ю. К., Хорев А. А.* Обоснование критериев эффективности защиты речевой информации от утечки по техническим каналам // Защита информации. INSIDE. – 2007. – № 2. – С. 18–25.
2. *Хорев А. А.* Оценка эффективности защиты информации от утечки по техническим каналам // Специальная техника, 2006. – № 6. – С. 53–61.
3. *Хорев А. А.* Оценка эффективности защиты информации от утечки по техническим каналам // Специальная техника, 2007. – № 1. – С. 51–64.
4. *Шеннон.* Связь при наличии шума. В кн. Работы по теории информации и кибернетике. – М.: Издательство иностранной литературы, 1963. – 827 с.
5. Засоби активного захисту мовної інформації з акустичними та віброакустичними джерелами випромінювання. Класифікація та загальні технічні вимоги. НД ТЗІ–Р–001–2000. ДСТСЗІ СБ України. – Київ.: – 2000. – 9 с.
6. НД ТЗІ 2.3-003-2001. Технічний захист мовної інформації в симетричних абонентських аналогових телефонних лініях. Засоби активного приховування мовної інформації. Генератори спеціальних сигналів. Методика випробувань. ДСТСЗІ СБ України. – Київ.: – 2001. – 21 с.
7. Советский энциклопедический словарь. / Гл. ред. А. М. Прохоров. 4-е изд. – М.: Сов. энциклопедия, 1989. – 1632 с.
8. *Отт Г.* Методы подавления шумов и помех в электронных системах. /пер с англ. – М.: Мир, 1978. – 317 с.
9. *Крауфорд Ф.* Волны. – М.: Наука, 1974. – 358 с.
10. Акустика: Справочник. Под ред. М.А. Сапожкова. – М.: Радио и связь, 1989. – 336 с.
11. Цифровая обработка сигналов / А. Б. Сергиенко – СПб.: Питер, 2003. – 608 с.
12. *Цвикер Э., Фельдкеллер Р.* Ухо как приемник информации. Пер. с нем. Под редакцией Б. Г. Белкина. – М.: Связь, 1971. – 255 с.
13. *Гайдышев И.* Анализ и обработка данных: специальный справочник. – СПб: Питер, 2001. – 752 с.
14. *Журавлев В. Н., Прокофьев М. И.* Анализ результатов артикуляционных и сегментальных испытаний сигналов маскирования речи. Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – 2006. – № 13. – С. 36–48.
15. ГОСТ Р 50840-95. Государственный стандарт Российской Федерации. Передача речи по трактам связи. Методы оценки качества, разборчивости и узнаваемости. Издание официальное. – М.: Госстандарт России, 1997. – 12 с.

Надійшла 3.10.07
Після доробки 17.10.07

В статті проведений аналіз методики цифрової кореляційної обробки контрольного фрагменту мови, що дозволяє на основі розрахунку коефіцієнта кореляції обґрунтувати аналітичну оцінку параметра якості передачі сигналу мови по каналам зв'язку та проводити аналіз ефективності адитивного маскування мовних сигналів.

The method of check utterance digital correlation processing, which allows to substantiate the communication path quality criteria analytic estimation on the base of correlation coefficient calculation and to analyze the speech signal additive masking effectiveness is under review.

УДК 004.75

М. Б. Ильяшенко

АЛГОРИТМ ОПТИМАЛЬНОГО РЕЗЕРВИРОВАНИЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ ПО КРИТЕРИЮ ЭФФЕКТИВНОГО ИСПОЛЬЗОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ УЗЛОВ

В работе предлагается алгоритм оптимального резервирования вычислительных ресурсов, способный осуществлять поиск областей резервирования, согласно критериям оптимальности по числу и максимальной загруженности задействованных вычислительных узлов. Приводится детальное описание алгоритма и набора предварительных условий, которые сужают область поиска переборной части алгоритма.

ВВЕДЕНИЕ

Алгоритмы управления распределенными вычислительными ресурсами, как правило, используют одну из двух возможных парадигм: централизованное управление ресурсами [1, 2] или контроль ресурсов локальными приложениями [3, 4]. Каждая парадигма имеет свои достоинства и недостатки, в частности, централи-

зованное управление ресурсами позволяет производить глобальную оптимизацию выделения, использования и переназначения ресурсов сети, но этот подход недостаточно масштабируем как в терминах производительности, так и устойчивости сети в целом. С другой стороны, второй подход, основанный на локальных менеджерах, управляющих ресурсами одного или нескольких компьютеров, может приводить к нестабильному выделению ресурсов и не позволяет получать оптимальные решения в масштабах всей сети по такому параметру, как например, средняя загруженность ресурсов сети. Каждый подход имеет свою область применения. Локальное управление ресурсами рационально использовать в сетях с частыми изменениями в структуре использования ресурсов и большим количеством мелких задача, выполняющихся в рамках сети GRID, в отличие от централизованного подхода, который лучше применять при небольшом количестве ресурсоемких задач, работающих в течение длительного времени и не требующих частого перераспределения ресурсов сети.

Кроме того, большинство подходов к распределению ресурсов сети не учитывают необходимость совместного выделения ресурсов (co-allocation), необходимого для исполнения параллельных задача в рамках сетей GRID. Наиболее важными видами ресурсов для параллельных задач являются продуктивность процессоров, объем доступной оперативной памяти и пропускная способность или латентность каналов связи для каждого вычислительного узла сети. Алгоритм глобального резервирования распределенных вычислительных ресурсов должен поддерживать все типы ресурсов, которые могут быть зарезервированы и при этом влияют на производительность программ. В рассмотренном алгоритме базовыми ресурсами, подлежащими резервированию, являются минимальная производительность процессов и минимальная необходимая полоса пропускания сети.

В работе использован графо-аналитический подход к решению задачи оптимального резервирования распределенных вычислительных ресурсов в сетях GRID. Недавний прогресс в развитии алгоритмов точной проверки графов на изоморфность [5–8] позволяет приступить к разработке новых алгоритмов глобальной оптимизации выделения ресурсов, способных работать с GRID сетями реальных размеров. Алгоритм оптимального резервирования распределенных вычислительных ресурсов, представленный в работе, основан на алгоритме нахождения граф-подграф изоморфизма на взвешенных графах [9] с модификациями, направленными на возможность распределения нескольких узлов параллельной задачи на один вычислительный узел сети.

1 ПОСТАНОВКА ЗАДАЧИ

Пусть даны два графа G_N и G_T. Пусть граф $G_N = (E_N, V_N, I_N, J_N)$ — представляет собой граф сети, где E_N — множество физических сетевых соединений (линий связи), V_N — множество вычислительных узлов сети (процессоров), I_N — множество весов, приписанных ребрам графа, представляющих собой пропускную способность сетевых соединений, измеренную в Mbps или Gbps и J_N — множество весов, приписанных вычислительным узлам графа сети, представляющих собой производительность процессоров, измеренную в MIPS или MFLOPS. Пусть граф $G_T = (E_T, V_T, I_T, J_T)$ — представляет собой граф параллельной задачи, для которой необходимо зарезервировать часть вычислительных ресурсов сети до начала вычислений, где E_T — множество ребер графа, соответствующих сетевым взаимодействиям между модулями параллельной программы, V_T — множество вершин, соответствующих модулям параллельной программы, I_T — множество весов, приписанных ребрам графа задачи, выражающих потребность в пропускной способности сети для данного сетевого взаимодействия, измеренную в Mbps или Gbps, J_T — множество весов, приписанных вершинная графа, соответствующих потребности модулей в вычислительной производительности процессорных узлов, измеренной в MIPS или MFLOPS.

В терминах определений, введенных выше, задача резервирования распределенных вычислительных ресурсов формализируется следующим образом. Необходимо найти подстановку $\varphi: V_T \to V_N$, такую, что для каждой пары вершин (ν_i, ν_j), если $(\nu_i, \nu_j) \in E_T \Rightarrow$ $\Rightarrow (\varphi(\nu_i), \varphi(\nu_j)) \in E_N$ и для $(i_i, i_j) \in I_T \le (\varphi(i_i), \varphi(i_j)) \in$ $\in I_N$, и $j_i \in J_T \le \varphi(j_i) \in J_N$.

Задача оптимизации резервирования ресурсов по критерию минимального количества задействованных вычислительных узлов сети означает, что среди всех $\varphi: V_T \to V_N$, выбирается та подстановка, при которой $|V_N| \to \min$.

2 АЛГОРИТМ РЕЗЕРВИРОВАНИЯ РЕСУРСОВ

2.1 Структура алгоритма

Как следует из постановки задачи, проблема схожа с задачей нахождения граф-подграф изоморфизма на взвешенных графах, стой лишь разницей, что допускается совмещение нескольких вершин графа задачи с одной и той же вершиной графа сети. Веса, приписанные вершинам и ребрам графа сети, являются ограничениями «сверху», налагаемыми на возможное решение.

Описание алгоритма удобно производить в терминах поиска в пространстве состояний. Каждому состоянию соответствует частичная подстановка $\varphi(s)$, содержащая лишь часть вершин графов, которые уже были совмещены.

Алгоритм состоит из предварительной и основной части.

2.2 Предварительная часть алгоритма

Основной задачей предварительной части алгоритма является выполнение всех проверок, которые основаны на данных известных до начала работы алгоритма и не базируются на информации о уже совмещенных вершинах частичной подстановки $\varphi(s)$. На их основе формируется матрица возможных совмещений. Так же в предварительной части алгоритма производится сортировка вершин графов.

Центральным элементом предварительной части алгоритма является матрица возможных совмещений. Это бинарная таблица $M_{i,j}$, каждая ячейка которой хранит агрегированное значение о возможности либо не возможности совмещения вершин $V_{N,i}$ и $V_{T,j}$. Значения матрицы формируются на основании предварительных проверок и могут принимать значения «истина» или «ложь», в зависимости от того перспективно или нет совмещение соответствующих вершин в основной части алгоритма на основании проверок, выполненных в предварительной части алгоритма.

В представленном алгоритме реализовано несколько базовых проверок, формирующих матрицу возможных совмещений $M_{i,j}$. Если результат проверки условия «истина», то значение матрицы, соответствующее проверяемым вершинам $V_{N,i}$ и $V_{T,j}$, устанавливается равным «ложь».

Пусть $|V_X|$ — степень вершины V_X. Тогда условие, основанное на сравнении степеней вершин, выглядит следующим образом:

$$|V_{N,i}| < |V_{T,j}| \Rightarrow M_{i,j} = \text{falce}.$$

Условие, основанное на сравнении весов, приписанных вершинам графов:

$$J_{N,i} < J_{T,j} \Rightarrow M_{i,j} = \text{false}.$$

Пусть $V\text{in}_X$ — подмножество, содержащее все вершины, связанные с вершиной V_X входящими ребрами, тогда условие, основанное на этом определении:

$$|V\text{in}_{N,i}| < |V\text{in}_{T,j}| \Rightarrow M_{i,j} = \text{false}.$$

Пусть $I\text{in}_X$ — подмножество весов, приписанных ребрам, входящим в вершину V_X. Условие, основанное на значении $I\text{in}_X$:

$$\sum I\text{in}_{N,i} < \sum I\text{in}_{T,j} \Rightarrow M_{i,j} = \text{false}.$$

Пусть $V\text{out}_X$ — подмножество, содержащее все вершины, связанные с вершиной V_X исходящими ребрами, тогда условие, основанное на этом определении:

$$|V\text{out}_{N,i}| < |V\text{out}_{T,i}| \Rightarrow M_{i,j} = \text{false}.$$

Пусть $I\text{out}_X$ — подмножество весов, приписанных ребрам, исходящим из вершины V_X. Условие, основанное на значении $I\text{out}_X$:

$$\sum I\text{out}_{N,i} < \sum I\text{out}_{T,j} \Rightarrow M_{i,j} = \text{false}.$$

Приведенный выше список простейших условий, не требующих сложных вычислений, тем не менее, часто приводит к значительному сокращению области поиска основной, переборной, части алгоритма. Но в дополнение к рассмотренным выше, предлагается использование условий, основанных на волновом разложении графов, более подробно описанном в работе [10]. Метод волнового разложения графов формирует много структурной информации о графе, которая может быть использована для построения более эффективных условий ограничения области поиска, используемых в предварительной части алгоритма. Наиболее важной частью волнового разложения графов, используемой в данном алгоритме, является подграф окружения вершины. Это подграф, состоящий из вершин, находящихся на расстоянии k ребер от исходной, где k — параметр, определяющий как много соседних вершин будет задействовано при формирования подграфа окружения. Другим важным параметром волнового разложения, использованным в работе, является число вершин, вошедших в каждую волну разложения графа, начиная с заданной вершины.

Пусть $W_{X,Y,k}$ — множество вершин, вошедших в волновое разложение графа X, начиная с вершины Y с максимальным реберным расстоянием до вершин k. Условие, основанное на этом определении:

$$|W_{N,i,k}| < |W_{T,j,k}| \Rightarrow M_{i,j} = \text{false},$$

где $k = 1 \ldots |W_{T,j}|$.

Пусть $U_{X,Y,k}$ — множество весов, соответствующих вершинам множества $W_{X,Y,k}$:

$$\sum U_{N,i,k} < \sum U_{T,j,k} \Rightarrow M_{i,j} = \text{false},$$

где $k = 1 \ldots |W_{T,j}|$.

Пусть $Q_{X,Y,k}$ — подмножество ребер, которые вошли в волновое разложение графа X, начиная с вершины Y с максимальным расстояние до вершин в k ребер и пусть $|Q_{X,Y,k}|$ — количество ребер в подмножестве $Q_{X,Y,k}$. Тогда формируется условие:

$$|Q_{N,i,k}| < |Q_{T,j,k}| \Rightarrow M_{i,j} = \text{false},$$

где $k = 1 \ldots |Q_{T,j}|$.

Пусть $R_{X,Y,k}$ — множество весов, приписанных ребрам, вошедшим в подмножество $Q_{X,Y,k}$:

$$\sum R_{N,i,k} < \sum R_{T,j,k} \Rightarrow M_{i,j} = \text{false},$$

где $k = 1 \ldots |Q_{T,j}|$.

Условия, основанные на волновом разложении графов, могут быть просчитаны для некоторого константного значения k или для всех возможных значений k. Во втором случае условия будут иметь большую вычислительную сложность, но так же лучше ограничивать область поиска переборной части алгоритма. Поскольку все условия предварительной части алгоритма имеют полиномиальную вычислительную сложность, то для больших графов имеет смысл выполнить все перечисленные выше условия. Некоторое дополнительное время, затраченное на условия предварительной части алгоритма, приведет к более значительному сокращению времени основной, переборной, не полиномиальной, части.

Еще одним важным действием, выполняемым в рамках предварительной части алгоритма, является сортировка вершин графов. Целью является перестановка вершин графов таким образом, чтобы получить более сильное ограничивающее условие, основанное на частичных подстановках, получаемых в процессе работы основной, переборной, части алгоритма. Идея основана на том, чтобы вначале совмещать вершины, имеющие больше внутренних реберных связей, и, как результат, формирующих более сильное ограничивающее условие переборной части. В программе сортируются только вершины графа G_T, описывающего требования параллельной задачи к распределенным ресурсам, в то время, как порядок следования вершин графа сети G_N остается без изменений, т. к. соответствие вершин графа G_N вершинам графа G_T будет устанавливаться уже в переборной части алгоритма.

Пусть $T_{T,i}$ — число ребер, инцидентных вершинам, имеющим индекс меньше, чем i, и пусть $P_{T,i} = \sum M_j$ — число возможных совмещений, доступных для вершины $V_{T,i}$ графа G_T. Тогда порядок следования вершин графа G_T определяется следующими выражениями:

$$V_{T,i} = V_{T,k},$$

где $T_{T,k} = \min(T_{T,j})$ для $j = (i+1) \ldots |V_T|$.

В случае, если $T_{T,i} = T_{T,k}$ применяется другое условие:

$$V_{T,i} = V_{T,k},$$

где $P_{T,k} = \min(P_{T,i}, P_{T,j})$.

2.3 Переборная часть алгоритма

Переборная часть алгоритма объединяет в себе все действия, направленные на поиск полной подстановки φ, являющейся решением задачи. Алгоритм представлен рекурсивной функцией поиска в пространстве состояний, которая на каждом шаге вложенности генерирует новую частичную постановку $\varphi_{i+1}(s)$ из предыдущей частичной подстановки $\varphi_i(s)$, путем добавления одной вершины в частичную подстановку.

Начальное состояние $\varphi_0(s) = 0$. На каждом шаге функция перебирает все вершины графа G_N, которые имеют пометку «истина» в строке матрицы возможных совмещений, соответствующей вершине $V_{T,i}$, используемой на текущем уровне вложенности алгоритма i. Для каждой вершины $V_{N,j}$, алгоритм производит несколько проверок, имеющих целью удостовериться, что все необходимые условия для включения новой вершины в частичную подстановку $\varphi_{i+1}(s)$ выполнены. Далее приведен полный список условий, используемых в переборной части алгоритма с более подробным описанием.

Наиболее важным является условие, основанное на значение соответствующего элемента матрицы возможных совмещений M:

$$M_{i,j} = \text{true}.$$

Пусть $T_{N,i}$ — число ребер, связывающих вершину $V_{N,i}$ с как минимум одной вершиной из вошедших в частичную подстановку $\varphi_i(s)$:

$$T_{N,i} > T_{N,j}.$$

Условие, основанное на множестве ребер, входящих в текущую частичную подстановку:

$$(\nu_i, \nu_k) = (\varphi_i(\nu_i), \varphi_i(\nu_k)),$$

где $k = 1 \ldots i$.

Условие, основанное на весах этих ребер:

$$(i_i, i_k) \geq (\varphi_i(i_i), \varphi_i(i_k)),$$

где $k = 1 \ldots i$.

Если все условия, описанные выше, истинны, то очередная пара вершин $(V_{T,i}, V_{N,j})$ будет добавлена в частичную подстановку $\varphi_i(s)$ и будет сформирована новая частичная подстановка $\varphi_{i+1}(s)$.

В этом списке последние два условия следуют из определения задачи и являются формальными строгими условиями, позволяющими очередной вершине войти в частичную подстановку. Когда эти условия последовательно применяются ко всем парам вершин, входящим в частичные подстановки $\varphi_i(s)$, для всех значений i от 1 до $|V_T|$, в результате формируется суммарное условие, прямо следующее из постановки задачи и гарантирующее, что полученная полная подстановка будет удовлетворять всем условиям, оговоренным в постановке задачи, т. е. будет являться решением задачи.

Перебор вариантов производится методом поиска в глубину.

2.4 Поиск оптимальной области резервирования

Для поиска оптимальной области резервирования используется особенность реализации алгоритма, заключающаяся в том, что перебор возможных подстановок $\varphi(s)$ можно продолжать, после нахождения первой полной подстановки и перебрать все полные подстановки, которых может быть несколько. Из всех полных подстановок, которые будут найдены алгоритмом, выбирается та, что соответствует критерию оптимальности.

Пусть подстановка $\varphi(s)$ ставит в соответствие вершинам графа задачи G_T часть вершин графа сети G_N: $\varphi(s): V_T \to V'_N \in V_N$.

Оптимальная область резервирования по критерию числа задействованных вычислительных узлов определяется следующим условием, налагаемым на полную подстановку:

$$\varphi(s): V_T \to V'_N \in V_N, \text{ где } |V'_N| \to \min.$$

Практический смысл оптимизации области резервирования по числу задействованных узлов заключается в том, что используется минимально возможное число физических компьютеров, что позволяет выключить остальные компьютеры или использовать их для совершенно других задач. Так же это упрощает администрирование, предоставляя возможность выдать права пользователям только на определенных компьютерах и передать минимально необходимое, но достаточное количество компьютеров в пользование или аренду.

Оптимальная область резервирования по критерию максимальной загруженности вычислительных узлов сети определяется следующим условием:

$$\varphi(s): V_T \to V'_N \in V_N,$$

при этом

$$\sum (J'_N - J_T) \to \min,$$

где J'_N — множество весов, приписанных вершинам графа сети V'_N, задействованным в полной подстановке.

Практический смысл оптимизации по критерию максимальной загруженности вычислительных узлов сети в наиболее полном использовании имеющихся вычислительных мощностей. Если такая оптимизация производится в масштабах всей сети для составного графа содержащего все задачи, выполняемые в сети, то как следствие получается глобальная оптимизация распределения зарезервированных областей, если же критерий оптимальности используется при добавлении каждой следующей задачи в сеть, то оптимизация зарезервированных областей производится жадным алгоритмом, который так же дает хорошие результаты оптимизации, без необходимости увеличения размерности задачи до масштабов всей сети.

ЗАКЛЮЧЕНИЕ

В работе представлен переборный алгоритм оптимизации резервирования распределенных вычислительных ресурсов по критериям числа и максимальной загруженности задействованных вычислительных узлов. В алгоритме использованы составные условия, ограничивающие область поиска переборной части алгоритма, в том числе, основанные на волновом разложении графов, значительно сокращающие время поиска оптимального решения.

Дальнейшие усилия будут приложены к оптимизации областей резервирования по критериям оптимального использования пропускной способности сети и возможности применения алгоритма для сверхбольших сетей, посредством разбиения задачи оптимизации на несколько подзадач, меньшей размерности.

ПЕРЕЧЕНЬ ССЫЛОК

1. *I. Foster, A. Roy, and V. Sander.* A quality of service architecture that combines resource reservation and application adaptation // Proceedings of the 8th International Workshop on Quality of Service (IWQOS). – Pittsburgh, PA. – June 2000. – P. 181–188.
2. *Sander V.* A Metacomputer Architecture Based on Cooperative Resource Man-agement // Proceedings of High Performance Computing and Networking Europe 1997 (HPCN 1997). – Wien. – April (1997). – P. 28–30.
3. *F. Berman, R. Wolski, S. Figueira, J. Schopf, and G. Shao.* Application level scheduling on distributed heterogeneous networks. // Proceedings of Supercomputing – 1996. – P. 39.
4. *Zeng Wandan, Chang Guiran, Zhang Dengke, Zheng Xiuying.* G-RSVPM: A Grid Resource Reservation Model // First International Conference on Semantics, Knowledge and Grid (SKG'05). – Guilin Guanxi, China – 2005. – P. 79.

5. *L. P. Cordella, P. Foggia, C. Sansone, M. Vento.* Performance evaluation of the VF Graph Matching Algoritmh // Proc. of the 10th ICIAP, IEEE Com-puter Society Press. – 1999. – P. 1172–1177.
6. *Bunke H., Vento M.* Benchmarking of graph matching algorithms. // Proceedings of the 2nd Workshop on Graph-based Representations. – Haindorf. – 1999. – P. 109–114.
7. *Cordella L. P., Foggia P., Sansone C., Vento M.* An improved algorithm for matching large graphs. // Proc. of the 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition. – Italy. – 2001. – P. 149–159.
8. *P. Foggia, C. Sansone, M. Vento.* A performance comparison of five algorithms for graph isomorphism. // Proc. of Ильяшенко М. Б. Разработка и исследование параллельного алгоритма проверки граф-подграф изоморфизма. // Радиоэлектроника. Информатика. Управление. – 2006. – № 1. – С. 63–69.
9. *Пинчук В. П.* Табличные инварианты на графах и их применение // Кибернетика и системный анализ. – 2001. – № 4. – С. 33–45.

Надійшла 20.07.07
Після доробки 26.10.07

В роботі запропоновано алгоритм оптимального резервування обчислювальних ресурсів, що здатен виконувати пошук областей резервування, згідно критеріям оптимальності по числу та максимальному завантаженню задіяних обчислювальних вузлів. Наводиться детальний опис алгоритму та набору попередніх умов, що звужують область пошуку у переборній частині алгоритму.

This paper presents algorithm for optimal distributed resources reservation, that allow to find reservation areas according to minimal amount of used computational nodes optimization criteria and maximal productivity usage of equipped computational nodes criteria. Paper contains detailed algorithm description, including set of preliminary conditions that reduce computational complexity of enumerating part of algorithm.

УДК:681.142.2

С. В. Курапов

КОНСТРУКТИВНЫЙ АЛГОРИТМ ДЛЯ РАСКРАСКИ КУБИЧЕСКИХ ГРАФОВ

В данной работе представлен конструктивный алгоритм для раскраски кубических графов с применением теории вращения вершин. Алгоритм основан на фундаментальной теореме Петерсена выделения 1-факторов и 2-факторов в кубических графах.

ВВЕДЕНИЕ

В 1913 г. Биркгоф ввел понятие неприводимого графа и доказал ряд теорем о свойствах таких графов. Пользуясь этими результатами, американский математик Франклин доказал, что гипотеза четырех красок верна для всех плоских графов с числом вершин до 31. Французский ученый Майер довел это число до 96. Хееш в 1969 г. свел вопрос о справедливости гипотезы четырех красок к исследованию достаточно большого так называемого неустранимого множества конфигураций. Хеешу удалось доказать, что после первого шага метода нейтрализации остается около 8900 положительных конфигураций, большинство из которых неприводимы [1]. В 1977 году доказательство гипотезы четырех красок было наконец получено К. Аппелем и У. Хакеном (Appel, Haken) последователями Хееша, и опубликовано в двух статьях [2].

Значительную часть рутинных проверок выполнил компьютер, и это революционное нововведение в сложившуюся практику дедуктивных рассуждений в чистой математике служит основанием для некоторого естественного скептицизма по отношению к данному доказательству, и по сей день.

Читатель данного доказательства, должен разобраться в 50 страницах текста и диаграмм, 85 страницах с почти 2500 дополнительными диаграммами, 400 страницами микрофишей, содержащими еще диаграммы, а также тысячи отдельных проверок утверждений, сделанных в 24 леммах основного текста. Вдобавок читатель узнает, что проверка некоторых фактов потребовала 1200 часов компьютерного времени, а при проверке вручную потребовалось бы гораздо больше. Статьи устрашающи по стилю и длине, и немногие математики прочли их сколько-нибудь подробно [3].

В литературе [4] приведено мнение большинства специалистов о том, что нет более быстрого метода раскрашивания карты, чем перебор всех вариантов. Там же приведена Фортран-программа на основе алгоритма полного перебора. Следует также заметить, что во всех этих подходах к решению проблемы, отсутствует понятие планарности, хотя гипотеза доказывается для плоских графов, отсутствует также и конструктивный алгоритм раскраски. Поэтому, исследования направленные на создание более эффективных алгоритмов раскраски карты актуальны.