УДК 519.171.1

ПОЛИНОМИАЛЬНЫЙ АЛГОРИТМ ВЫЧИСЛЕНИЯ ПОЛНОГО ИНВАРИАНТА ГРАФА НА ОСНОВЕ ИНТЕГРАЛЬНОГО ОПИСАТЕЛЯ СТРУКТУРЫ

В.К. Погребной, Ан.В. Погребной

Томский политехнический университет
E-mail: avpogrebnoy@gmail.com

Актуальность исследования *заключается в том, что проблема поиска полного инварианта графа и полиномиального алгоритма его вычисления остаётся нерешенной.*

Цель работы *состоит в нахождении полного инварианта обыкновенного графа на основе интегрального описателя абстрактной структуры и в разработке эффективного алгоритма вычисления полного инварианта.*

Методы исследования *базируются на теории графов и теории интеграции кодов структурных различий в абстрактных структурах графов.*

В результате исследований *предложен алгоритм решения одной из наиболее сложных задач теории графов – вычисление полного инварианта графа. Алгоритм основан на методах свободной и зависимой интеграции кодов структурных различий в графе и характеризуется простотой, эффективностью, и имеет полиномиальную оценку предельного объема вычислений. Полный инвариант представлен в виде вектора интегральных описателей вершин абстрактной структуры графа и содержит информацию для формирования подстановки изоморфизма. На языке Java разработано программное средство GraphISD, реализующее предложенный алгоритм. Приведены примеры вычисления полных инвариантов при свободной и зависимой интеграции.*

Ключевые слова:
Полный инвариант графа, изоморфизм графов, интегральный описатель структуры, абстрактная структура графа, область интеграции кодов, полиномиальность алгоритма.

Содержательная формулировка проблемы

В теории графов под инвариантом понимают некоторую количественную меру, которая характеризует структуру графа и не зависит от нумерации его вершин [1]. Например, упорядоченный по возрастанию или убыванию вектор значений степеней вершин графа является его инвариантом. Очевидно, что для двух изоморфных графов G и H инварианты, сформированные на основе степеней вершин, обозначим их $S(G)$ и $S(H)$, совпадают, т. е. $S(G)=S(H)$. К настоящему времени накоплено много типов инвариантов, каждый из которых отражает некоторые особенности структуры графа. Большое число инвариантов (топологических индексов) введено для исследования структуры химических графов [2]. В частности, для взвешенных обыкновенных графов авторами в работе [3] введен ряд инвариантов на основе оценок компактности, а также на основе компактных подграфов.

Естественным является стремление найти инвариант или комбинацию инвариантов, которые бы однозначно представляли структуру графа с точностью до изоморфизма. Такие инварианты называются полными. Равенство полных инвариантов у двух графов является необходимым и достаточным условием их изоморфизма. В работе [4] подмечено, что попытку получить однозначное представление структуры с помощью инвариантов, характеризующих её отдельные свойства, можно сравнить с попыткой получить портрет лица путем составления фоторобота, используя при этом совокупность некоторых признаков. Как в первом случае, так и во втором, удаётся лишь достигнуть некоторого приближения к реальной структуре графа и к портрету конкретного лица.

Основная проблема здесь заключается в том, что в структуре графа множество количественных оценок тесно связано с множеством скрытых отношений между ними. Поэтому попытки охватить всё многообразие этих отношений с помощью некоторой количественной меры, способной однозначно представить структуру графа, в общем случае приводят к неполиномиальным алгоритмам, сравнимым по сложности непосредственно с самой задачей определения изоморфизма.

В статье предлагается изменить подход к решению данной проблемы, отойти от поиска полного инварианта в виде некоторой количественной меры и сосредоточиться на получении полного инварианта на основе интегрального описателя структуры (Integral Structure Descriptor – ISD), используя для этого метод интеграции структурных различий (метод ISD), изложенный в [4].

В методе ISD для всех вершин графа рекуррентно формируется иерархическая система числовых кодов, которая однозначно представляет абстрактную структуру графа. На верхнем уровне этой системы каждой вершине графа сопоставляется уникальный код ISD, который интегрально описывает положение данной вершины относительно всех других вершин графа. Значения числовых кодов при этом не несут никакой смысловой нагрузки, а выступают в роли описателей (дескрипторов), идентифицирующих структуру графа.

Полный инвариант, вычисленный с помощью метода ISD, зависит от используемой при формировании числовых кодов системы кодирования, которая в [4] названа областью интеграции кодов. Поэтому для сравнения полных инвариантов графов G и H необходимо, чтобы коды ISD были получены относительно одной области интеграции. В этом случае равенство полных инвариантов гарантирует изоморфизм графов G и H.

Очевидно, что равенство полных инвариантов имеет смысл проверять для множества графов, у которых некоторые заранее заданные признаки (инварианты) совпадают. Например, в качестве такого множества, обозначим его $M(S)$, примем графы с равными степенными инвариантами, представленными заданным вектором S. Полные инварианты графов множества $M(S)$ разбивают его на классы. Каждый класс содержит изоморфные графы, т. к. у них полные инварианты совпадают.

Учитывая, что класс содержит изоморфные графы, любой из них, например граф G, может представлять данный класс. В этом случае класс обозначается как $G(S)$. Для графа G, как представителя класса $G(S)$ определяется полный инвариант $P(G)$, который в последующем рассматривается в роли образа (эталона) при поиске других графов из множества $M(S)$, относящихся к классу $G(S)$. Следуя концепции предложенного в [4] метода ISD, в ходе определения $P(G)$ формируется область интеграции кодов $W_G(S)$, которая наряду с $P(G)$ применяется в качестве эталона и устанавливает правила кодирования структурных различий для графов класса $G(S)$. Тогда полный инвариант $P_G(H)$ графа H из множества $M(S)$, вычисленный в соответствии с системой кодирования области $W_G(S)$, можно сравнивать с $P(G)$. Если при этом $P_G(H)=P(G)$, то граф H изоморфен графу G и, следовательно, принадлежит классу $G(S)$.

Формирование области интеграции $W_G(S)$ и определение полного инварианта $P(G)$

Для задания множества графов можно использовать любые тривиально вычисляемые инварианты, например упомянутый выше степенной инвариант S. Все графы, у которых этот инвариант совпадает, образуют соответствующие множества $M(S)$. Будем считать, что степенной инвариант в виде вектора S задан и определяет множество связных обыкновенных графов $M(S)$. В множестве $M(S)$ произвольно выберем один из графов, например G, который будет представлять класс $G(S)$. Формирование области интеграции $W_G(S)$ для класса графов $G(S)$, представленного графом G со степенным инвариантом S, производится с применением метода свободной интеграции кодов структурных различий [4]. Граф $G=(E,U)$ представим матрицей смежности вершин $A=\|a_{ij}\|_{n\times n}$, $a_{ij}=1$, если вершины e_i и e_j связаны ребром u_{ij}, $a_{ij}=0$, в противном случае.

Введём n-мерный вектор $D^k=\{d_i^k\}$, элементы d_i^k которого будут соответствовать кодовым числам интегральных описателей вершин e_i на k-м шаге интеграции кодов структурных различий графа G. Если значение элемента d_i^k в векторе D^k встречается один раз, т. е. не повторяется, то такой элемент будем обозначать d_{i*}^k. На шаге $k=0$ все элементы d_i^0 вектора D^0 принимаются равными 1. Метод свободной интеграции последовательно выполняет преобразования вектора D^k в D^{k+1} до тех пор, пока на некотором шаге k не будет получен вектор $D^{k+1}=D$, у которого все элементы d_i^{k+1} окажутся обозначены как d_{i*}^{k+1}. Используемая при этом система кодирования оформляется в виде области интеграции $W_G(S)$ для класса $G(S)$, представляемого графом G.

Алгоритм свободной интеграции для формирования области $W_G(S)$ и определения полного инварианта $P(G)$ включает следующие операции.

1. Используя операцию попарного произведения элементов двух векторов, обозначим её символом $\otimes$, для каждой вектор-строки A_i (вектор-столбца A_j) выполним $A_i\otimes D^k=Z_i(D^k)=(a_{i1}d_1^k,a_{i2}d_2^k,\ldots,a_{in}d_n^k)$. Данная операция не выполняется для A_j, которым соответствуют элементы d_{i*}^k, внесённые в вектор D^k как виртуальные различия (см. п. 3). Нулевые значения элементов вектора $Z_i(D^k)$ исключаются, а множество ненулевых упорядочивается по возрастанию значений и обозначается Z_i^k. В результате имеем совокупность множеств $Z^k=\{Z_i^k\}$.
2. Кодовые числа d_{i*}^k, помеченные в D^k как виртуальные различия, переносятся в D^{k+1} без изменения значений. Далее для множеств $Z_i^k\in Z^k$ назначаются кодовые числа d_i^{k+1}, которые включаются в вектор D^{k+1}. Выбор кодового числа d_i^{k+1} при очередном назначении осуществляется из общего множества чисел $I=(1,2,\ldots,n)$. Назначение кодовых чисел выполняется таким образом, чтобы разные множества Z_i^k получили разные кодовые числа d_i^{k+1}, т. е.

$$\forall(Z_i^k,Z_j^k)\subset Z^k[(Z_i^k\neq Z_j^k)\,\&\,(i\neq j)]\Rightarrow(d_i^{k+1}\neq d_j^{k+1}).$$

Для этого последовательно анализируется каждое множество $Z_i^k\in Z^k$. Если Z_i^k совпадает с одним из множеств $Z_j^k\in Z^k$, для которого уже назначен код d_j^{k+1}, то множеству Z_i^k назначается код $d_i^{k+1}=d_j^{k+1}$. Если Z_i^k отличается от множеств Z_j^k с назначенными кодами d_j^{k+1}, то множеству Z_i^k назначается код d_i^{k+1} по правилу:

$$d_i^{k+1}=\min(I\setminus\tilde{I}(D^{k+1})). \qquad (1)$$

Здесь $\tilde{I}(D^{k+1})$ – множество элементов d_i^{k+1} внесённых ранее в вектор D^{k+1} на данном шаге интеграции. Код d_i^{k+1} заносится в вектор D^{k+1} и анализ очередного множества Z_i^k повторяется. Процесс заканчивается, когда всем Z_i^k будут назначены коды d_i^{k+1}. Множество кодовых чисел в векторе D^{k+1} обозначим $I(D^{k+1})$.

3. Выполняется анализ вектора D^{k+1}. Если $\max(I(D^{k+1}))>\max(I(D^k))$, то проверяется наличие d_i^{k+1}, которые в векторе D^{k+1} встречаются один раз, и они помечаются кодом d_{i*}^{k+1}. Если $\max(I(D^{k+1}))=\max(I(D^k))$, то в векторе D^{k+1} выбирается группа элементов с одинаковыми значениями d_i^{k+1} и с минимальным числом элементов в этой группе. Один из элементов в выбранной группе заменяется на элемент d_{i*}^{k+1} по правилу:

$$d_{i*}^{k+1}=\max(I(d_i^{k+1}))+1. \qquad (2)$$

Далее выполняются действия, как и в случае соблюдения условия $\max(I(D^{k+1}))>\max(I(D^k))$. Назначение d_{i*}^{k+1} по правилу (2) можно рассматривать как искусственное внесение в граф виртуального различия.

4. Область интеграции для класса графов $G(S)$, обозначенную как $W_G(S)$, представим совокупностью отдельных записей, сформированных для каждого k-го шага интеграции, в виде:

$$k=(k,k_v):\{n_d(Z_i^k)\Rightarrow d_i^{k+1}\};\{(Z_i^k)\Rightarrow d_{i*}^{k+1}\}. \quad (3)$$

В записи (3) указывается шаг, равный k или k_v, если на k-м шаге интеграции использовалось очередное v-е, $v=1,2,...,V$, назначение кода d_{i*}^{k+1} по правилу (2). Множество $\{n_d(Z_i^k)\Rightarrow d_i^{k+1}\}$ отражает назначение кода d_i^{k+1} по правилу (1), и при этом число множеств Z_i^k с одинаковым кодом d оказалось равным n_d. Множество $\{(Z_i^k)\Rightarrow d_{i*}^{k+1}\}$ отражает уникальные коды d_{i*}^{k+1}, назначенные по правилу (1) или (2).

5. Операции 1–4 соответствуют выполнению одного k-го шага свободной интеграции кодов по преобразованию векторов $D^k\Rightarrow D^{k+1}$ и k-й записи в область $W_G(S)$. Выполнение шагов повторяется для $k=0,1,2,...\Delta k,\Delta k\le n$, и заканчивается, когда в векторе D^{k+1} все элементы окажутся уникальными кодами d_{i*}^{k+1}. Данный вектор является интегральным описателем структуры (ISD) графа G и в последующем обозначается как $D(G)=\{d_i\}$, $i=1,2,...,n$.
6. На основе вектора $D(G)$ для графа G вычисляется полный инвариант $P(G)=\{P_i(G)\}$. С этой целью для каждого d_i над вектор-строкой A_i графа G и вектором $D(G)$ выполняется операция попарного произведения: $A_i\otimes D(G)=\tilde{D}_i(G)$. После исключения из вектора $\tilde{D}_i(G)$ нулевых элементов и упорядочения по возрастанию ненулевых значений кодов d_i получим множество Z_i, которое, по существу, является инцидентором $F(d_i)$ вершины абстрактной структуры графа G, поименованной уникальным кодом d_i. Запись в виде $d_i(F(d_i))$ принимается в качестве элемента $\tilde{P}_i(G)$ вектора $\tilde{P}(G)$. Упорядоченная по возрастанию значений кодов d_i последовательность элементов $d_i(F(d_i))$ в n-мерном векторе $\tilde{P}(G)$ является полным инвариантом $P(G)=\{d_i(F(d_i))\}$ графа G и представляет абстрактную структуру графа, которая не зависит от исходной нумерации его вершин.

Последовательность действий при получении $P(G)$ представим в виде:

$$A_i\otimes D(G)=\tilde{D}_i(G);\ \tilde{D}_i(G)\Rightarrow Z_i=F(d_i);$$

$$d_i(F(d_i))\Rightarrow\tilde{P}_i(G);\ \{\tilde{P}_i(G)\}=\tilde{P}(G)\Rightarrow P(G). \quad (4)$$

Заметим, что элементы $P_i(G)$ вектора $P(G)$ могут содержать только инциденторы $F(d_i)$, а числовые коды d_i вершин абстрактной структуры графа G становятся избыточными, т. к. значения d_i изменяются от 1 до n и дублируют порядковые номера элементов в векторе $P(G)$.

В данном алгоритме коды D_i^k множествам Z_i^k назначались свободно, можно сказать произвольно, но в рамках заданных правил. Поэтому полученную систему кодирования, представленную областью $W_G(S)$, следует рассматривать как одну из многих возможных систем. При этом проблем с неоднозначностью кодирования здесь не возникает, т. к. система кодирования работает с множеством графов $M(S)$, замкнутым относительно инварианта S, и для любого графа $H\in M(S)$ проверка наличия полного инварианта $P_G(H)=P(G)$ осуществляется в зависимости от $W_G(S)$. Если найти $P_G(H)=P(G)$ не удалось, то граф H потенциально может стать представителем другого класса $H(S)$ с соответствующей областью $W_H(S)$. В этом случае принадлежность любого графа $Q\in M(S)\backslash(G,H)$ к классам $G(S)$ или $H(S)$ будет определяться наличием полного инварианта $P_G(Q)$ или $P_H(Q)$, вычисляемого относительно области $W_G(S)$ или $W_H(S)$.

Проверка наличия полного инварианта $P_G(H)$ относительно области $W_G(S)$

В предыдущем разделе показано, как для графа G из множества $M(S)$ с помощью метода ISD в условиях свободной интеграции кодов можно получить полный инвариант $P(G)$ и область $W_G(S)$. При наличии $W_G(S)$, представляющей класс $G(S)$, имеется возможность проверить принадлежность любого из графов $M(S)\backslash G$ к классу $G(S)$. Очевидно, что граф $H\in M(S)\backslash G$ принадлежит классу $G(S)$, если его полный инвариант $P_G(H)$, вычисленный с использованием системы кодирования $W_G(S)$, будет равен $P(G)$. Полный инвариант $P_G(H)$, полученный в системе $W_G(S)$ и равный $P(G)$, помечается индексом G, подчеркивая тем самым зависимое назначение кодов относительно области $W_G(S)$. Для проверки наличия у графа H полного инварианта $P_G(H)$ ниже предлагается алгоритм, использующий метод ISD с зависимой интеграцией кодов.

Алгоритм поиска $P_G(H)$ с использованием зависимой интеграции, в сравнении с алгоритмом определения $P(G)$ на основе свободной интеграции, имеет два существенных отличия. Первое из них – это назначение кодов d_i^{k+1} множествам Z_i^k в строгом соответствии с системой кодирования $W_G(S)$. Второе отличие связано с изменением правила «разрушения» устойчивых групп. Если при свободной интеграции виртуальное различие вводилось для одной (любой) вершины устойчивой группы, то при зависимой интеграции все вершины устойчивой группы последовательно принимаются в качестве претендентов для введения виртуального различия.

Напомним, что при свободной интеграции для введения виртуального различия выбирается устойчивая группа с меньшим числом вершин. Такую группу обозначим $E_d^{k_v}(G)$. Индекс k_v, $v=1,2,...,V$, обозначает v-й порядковый номер k-го шага, на котором вектор $D^{k_v}(G)$ содержит устойчивые группы. При зависимой интеграции соответствующую устойчивую группу в векторе $D^{k_v}(H)$ будем обозначать $E_d^{k_v}(H)$.

Алгоритм зависимой интеграции для поиска полного инварианта $P_G(H)$ включает следующие операции.

1. Формирование множеств $Z^k=\{Z_j^k\}$ осуществляется в полном соответствии с 1-й операцией алгоритма свободной интеграции.
2. Уникальные коды d_{j*}^k, соответствующие виртуальным различиям, переносятся в вектор $D^{k+1}(H)$. Назначение кодов d_j^{k+1} множествам $\{Z_j^k\}$ производится путём копирования кодов, принятых в k-й строке системы кодирования $W_G(S)$. Если соответствие между всеми $\{Z_j^k\}$ и $W_G(S)$ достигнуто, а вектор $D^{k+1}(H)\neq D(H)$, то относительно $D^{k+1}(H)$ выполняются операции 1 и 2. При условии, что $D^{k+1}(H)=D(H)$, т. е. $D^{k+1}(H)$ содержит только d_{j*}^{k+1}, алгоритм переходит к выполнению операции 4.

Отсутствие соответствия между Z_j^k и $W_G(S)$ означает, что вектор $D^{k+1}(H)$ получить не удалось и процесс зависимой интеграции на k-м шаге прерывается. Далее выполняется операция 3.

3. Процесс зависимой интеграции делает «обратный ход». Для этого в цепи переходов от $D^0(H)$ к $D^k(H)$ производится поиск вектора $D^{k_v}(H)$ с наибольшим значением индекса v, т. е. последний $D^{k_v}(H)$ в цепи переходов. При этом возможны три ситуации:
   а) Вектор $D^{k_v}(H)$ не найден, т. е. обратный ход заканчивается на векторе $D^0(H)$. Это означает, что граф H не имеет полного инварианта $P_G(H)$, т. е. не принадлежит классу $G(S)$. В этом случае алгоритм заканчивает работу.
   б) Вектор $D^{k_v}(H)$ найден, но в устойчивой группе $D_d^{k_v}(H)=D_d^{k_v}(G)$ всем вершинам $e_j\in D_d^{k_v}(H)$ виртуальные различия уже назначались. В этом случае выполнение операции 3 повторяется, т. е. обратный ход продолжается с целью поиска вектора $D^{k_v}(H)$.
   в) Вектор $D^{k_v}(H)$ найден и в группе $D_d^{k_v}(H)$ имеется вершина e_j, которой виртуальное различие не назначалось. Вершине $e_j\in D_d^{k_v}(H)$ согласно $W_G(S)$ назначается виртуальное различие, и процесс интеграции продолжает работу с вектором $D^{k_v}(H)$, т. е. выполняются операции 1 и 2.
4. Для вектора $D_G(H)$ вычисляется полный инвариант $P_G(H)$. При этом последовательность действий соответствует записи (4), и выполняются они аналогично операции (6) в алгоритме свободной интеграции для графа G. На этом алгоритм заканчивает работу.

С помощью данного алгоритма вычисляется полный инвариант $P_G(H)$ при условии, что он существует для графа H. Полный инвариант $P_G(H)$ соответствует определенной подстановке изоморфизма графов G и H. В общем случае для этих графов могут существовать и другие варианты подстановок изоморфизма. Если выделение таких подстановок актуально, то после вычисления $P_G(H)$ следует перейти к операции 3, делая «обратный ход» уже в цепи переходов от $D^0(H)$ к $D^{k+1}(H)=D_G(H)$.

Примеры схем поиска полных инвариантов при свободной и зависимой интеграции кодов структурных различий приведен на рис. 1. На рис. 1, *а* показана цепь переходов для графа G от вектора $D^0(G)$ до вектора $D(G)$ и соответствующего полного инварианта $P(G)$. В цепи выделены только векторы D^{k_v} и устойчивые группы $E_d^{k_v}(G)$, выбранные для введения виртуальных различий. Принятая при этом система кодирования отражается в $W_G(S)$.

Схема поиска $P_G(H)$ относительно $W_G(S)$ показана на рис. 1, *б*. Здесь короткие дуги, выходящие из вершин устойчивых групп, указывают, что введенное виртуальное различие не привело к успеху, т. е. возникло несоответствие между Z_j^k и $W_G(S)$ и потребовался «обратный ход». Длинные дуги означают, что очередной вектор $D^{k_v}(H)$ или $D_G(H)$ достигнуть удалось.

Рис. 1, *в* отражает выполнение «обратного хода» для всех неуспешных и успешных продвижений процесса интеграции (короткие и длинные дуги). В данном случае схема отражает перебор всех цепей переходов относительно системы кодирования $W_G(S)$. Такой перебор соответствует предельному объему вычислений при поиске интегрального описателя $D_G(H)$ и соответствующего полного инварианта $P_G(H)$. Полиномиальность оценки сложности данного перебора показана в [6] и легко подтверждается примером на рис. 1, *в*.

Из рис. 1, *в* следует, что в графе H относительно $W_G(S)$ полный инвариант $P_G(H)$ найден для трёх вариантов подстановок изоморфизма (автоморфизмов), соответствующих интегральным описателям $D_G^a(H)$, $D_G^b(H)$, $D_G^c(H)$. В случае, если граф $H\notin G(S)$, т. е. для него не существует $P_G(H)=P(G)$, схему поиска $P_G(H)$ с перебором всех цепей, аналогичную рис. 1, *в*, можно использовать для анализа сходства графов G и H. Изучение этих возможностей требует отдельных исследований. Некоторые вопросы по проблеме сходства двух графов на основе метода ISD были рассмотрены в работе [5].

Примеры получения области интеграции и полных инвариантов

Работу алгоритма формирования области интеграции $W_G(S)$ и получения полного инварианта $P(G)$ покажем на примере однородного графа G из класса графов $G(S)$ с $n=10$ и степенью вершин $S=3$. Для получения полного инварианта однородные графы представляют наибольшую сложность, поэтому в качестве примера выбран граф данного класса. В верхней части рис. 2 представлен граф G в виде матрицы A и процесс получения вектора $D(G)$ и данных для формирования записей (3) в область интеграции кодов $W_G(S)$.

Элементы в множествах Z_i^k на рис. 2 не разделены запятыми. В данном случае это не вносит путаницу, т. к. все элементы являются одноразрядными числами. Для получения вектора $D(G)=D^{k=8}$ потребовалось 8 шагов интеграции кодов. Правило (2) применялось дважды – для вектора D^{1_1} (назначен код $d_{1*}^1=2$) и для вектора D^{6_2} (назначен код $d_{2*}^6=8$). Область интеграции $W_G(S)$ для графов рас-

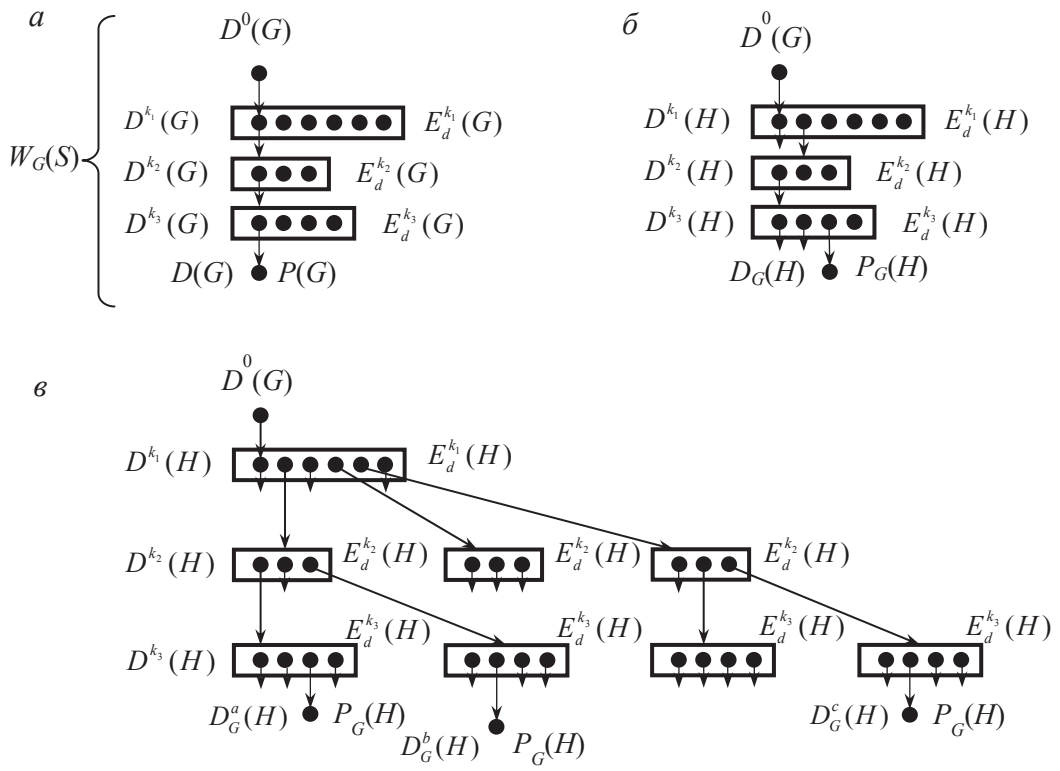

Рис. 1. *Схемы поиска полных инвариантов при свободной и зависимой интеграции*

сматриваемого класса $G(S)$, полученная на примере графа G, имеет следующий вид:

$k_1 = 0:\ 10(1,1,1) \Rightarrow 1;$

$k_1 = 1:\ 9(1,1,1) \Rightarrow 1; (1,1,1) \Rightarrow 2^*;$

$k = 2:\ 3(1,1,2) \Rightarrow 1;\ 6(1,1,1) \Rightarrow 3;$

$k = 3:\ 2(1,2,3) \Rightarrow 1;\ 4(1,3,3) \Rightarrow 3;$
$2(3,3,3) \Rightarrow 4;\ (2,3,3) \Rightarrow 5;$

$k = 4:\ 2(1,2,3) \Rightarrow 1;\ 2(1,3,4) \Rightarrow 3;$
$2(3,3,4) \Rightarrow 4;\ (2,3,3) \Rightarrow 5;\ 2(3,4,5) \Rightarrow 6;$

$k = 5:\ 2(1,2,3) \Rightarrow 1;\ 2(1,4,6) \Rightarrow 3;\ (3,5,4) \Rightarrow 4;$
$(2,6,6) \Rightarrow 5;\ 2(3,4,5) \Rightarrow 6;\ (4,6,6) \Rightarrow 7;$

$k_2 = 6:\ (1,2,3) \Rightarrow 1;\ 2(1,4,6) \Rightarrow 3;$
$(3,3,7) \Rightarrow 4;\ (2,6,6) \Rightarrow 5;\ 2(3,5,7) \Rightarrow 6;$
$(4,6,6) \Rightarrow 7;\ (1,2,3) \Rightarrow 8^*;$

$k = 7:\ (4,6,8) \Rightarrow 1;\ (3,3,7) \Rightarrow 3;\ (1,4,6) \Rightarrow 4;$
$(2,3,8) \Rightarrow 5;\ (2,6,6) \Rightarrow 6;\ 2(3,5,7) \Rightarrow 7;$
$(4,6,6) \Rightarrow 9;$

$k = 8:\ (3,7,8) \Rightarrow 1;\ (1,4,9) \Rightarrow 3;\ (3,5,7) \Rightarrow 4;$
$(2,4,8) \Rightarrow 5;\ (2,7,7) \Rightarrow 6;\ (1,6,9) \Rightarrow 7;$
$(3,7,7) \Rightarrow 9;\ (4,6,9) \Rightarrow 10.$

После вычисления согласно (4) вектора $\tilde{P}(G)=\{d_i(F(d_i))\}$, упорядочения в нём элементов $d_i(F(d_i))$ относительно значений кодов d_i получим полный инвариант графа G в виде: $P(G)$=(1(3,7,8), 2(5,6,8), 3(1,4,9), 4(3,5,10), 5(2,4,8), 6(2,7,10), 7(1,6,9), 8(1,2,5), 9(3,7,10), 10(4,6,9)).

Область интегрции не зависит от нумерации вершин и представляет собой иерархическую систему кодирования, которая используется в качестве эталона при выполнении алгоритма зависимой интеграции кодов для всех графов рассматриваемого класса. При этом для графов, изоморфных графу G, этот эталон соблюдается, а для неизоморфных – нет. Покажем это на примере двух графов H и Q, и определим их принадлежность классу $G(S)$, представленному графом G. В данном случае назначение кодов в ходе работы алгоритма интеграции должно производиться не по правилам (1) и (2), а в строгом соответствии с системой кодирования области интеграции $W_G(S)$, полученной ранее для заданного класса $G(S)$.

Для графа H (средняя часть рис. 2) после назначения по правилу (2) в вектор D^{1_1} кода d^1_{1*}=2 уже на 3-м шаге интеграции обнаружилось структурное различие, которое не соответствует записи в $W_G(S)$. Вторая попытка с назначением d^1_{2*} оказалась успешной, т. е. процесс интеграции прошел в соответствии с $W_G(S)$, в том числе и с учётом назначения по правилу (2) кода d^6_{2*}=8. Полный инвариант $P_G(H)$, полученный на основе вектора $D_G(H)$, в соответствии с правилами (4) совпадает с $P(G)$. Из этого следует, что граф H изоморфен графу G, а сопоставляя в векторах $D(G)$ и $D_G(H)$ коды d_i, получим подстановку изоморфизма: 1(3,9), 2(1,2), 3(4,10),

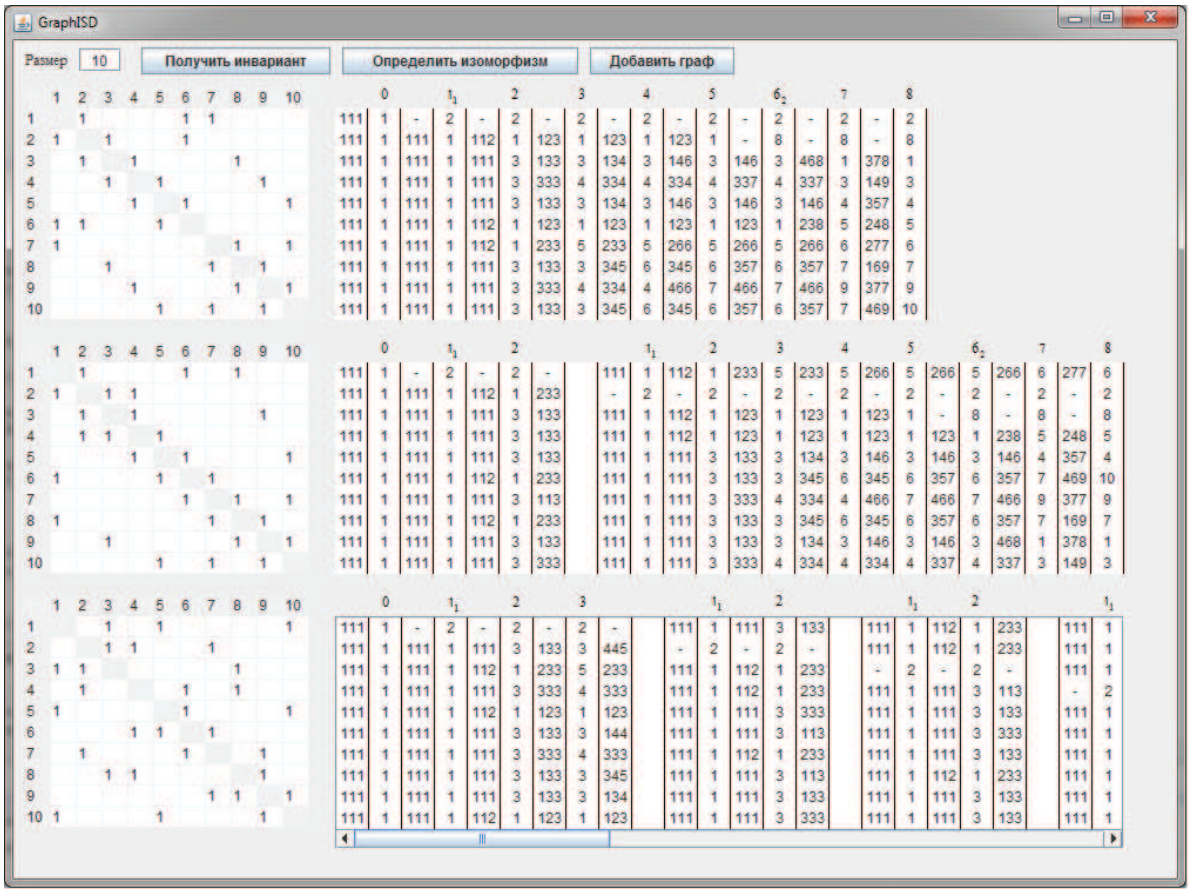

Рис. 2. *Примеры работы алгоритмов свободной и зависимой интеграции кодов структурных различий*

4(5,5), 5(6,4), 6(7,1), 7(8,8), 8(2,3), 9(9,7), 10(10,6). В подстановке перед скобками указаны значения кодов d_i, в скобках на первом месте указан номер вершины графа G, а на втором – графа H.

На рис. 3 представлена абстрактная структура графов G и H, построенная на основе инцидентов $F(d_i)$, содержащихся в полном инварианте $P(G)=P_G(H)$. Вершины данной структуры обозначены кодами d_i с указанием соответствующих пар вершин подстановки.

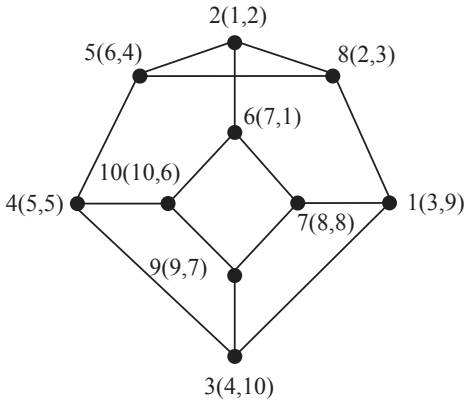

Рис. 3. *Абстрактная структура графов G и H*

Заметим, что на основе полного инварианта может быть построена только абстрактная структура графа, а для установления соответствия между вершинами структуры и вершинами графа необходимо использовать векторы $D(G)$ и $D_G(H)$ или $\tilde{P}(G)$ и $\tilde{P}_G(G)$.

Для графа Q (нижняя часть рис. 2) попытки получить полный инвариант $P_G(Q)=P(G)$ путём интеграции кодов относительно области $W_G(S)$ не привели к успеху. На рис. 2 приведены 3 попытки назначения кодов $d^1_{1*}=2$, $d^1_{2*}=2$, $d^1_{3*}=2$. В первой из них структурные различия обнаружены на шаге $k=4$, а в двух последних – на шаге $k=3$. Все последующие попытки назначения кодов $d^1_{i*}=2$, $i=4,5,\ldots,10$ (на рис. 2 не показаны) также не привели к успеху. В частности, для $d^1_{10*}=2$ структурные различия обнаружились только на шаге $k=5$. Из этого следует, что у графа Q отсутствует полный инвариант $P_G(Q)=P(G)$, т. е. граф Q неизоморфен графам G и H. Для графа Q, как и для графа G, может быть получен инвариант $P(Q)\neq P(G)$ с соответствующей областью интеграции $W_Q(S)$. В классе $Q(S)$ поиск графов с полными инвариантами, равными $P(Q)$, следует вести относительно области $W_Q(S)$.

Заключение

В статье предложен алгоритм решения одной из наиболее сложных задач теории графов – вычисление полного инварианта графа. Алгоритм характеризуется простотой, высокой эффективностью и имеет полиномиальную оценку предельного объема вычислений. Полный инвариант удалось получить на основе интегрального описателя структуры графа, и поэтому вопросы поиска и существования алгоритма вычисления полного инварианта в виде количественной меры остаются открытыми. Результаты исследований, полученные в данной работе, по-видимому, добавят оптимизма, а возможно и окажут помощь в поиске ответов на эти вопросы. Однако здесь необходимо иметь ввиду следующее обстоятельство. Если полные инварианты в виде количественной меры будут способны лишь определять наличие изоморфизма, то ценность таких инвариантов невелика, т. к. изоморфизм легко определяется методом интеграции структурных различий. Дело в том, что при получении полных инвариантов на основе интегральных описателей ценными являются не столько сами инварианты, сколько соответствующие процессы интеграции кодов структурных различий. При этом непосредственно инварианты определяют наличие изоморфизма, выступают в роли описателей абстрактной структуры графов, содержат информацию о подстановке изоморфизма, а по процессам интеграции можно судить о сходстве графов на каждом шаге интеграции, выделять и при необходимости интерпретировать структурные различия, однородные и устойчивые группы, анализировать другие, сопутствующие процессу интеграции, признаки.

Работа выполнена в рамках государственного задания «Наука».

СПИСОК ЛИТЕРАТУРЫ

1. Зыков А.А. Основы теории графов. – М.: Вузовская книга, 2004. – 664 с.
2. Дэмер М., Эммерт-Штрайб Ф., Цой Ю. Р., Вармуза К. Новый функционал информативности для анализа структуры химических графов // Известия Томского политехнического университета. – 2010. – Т. 316. – № 5. – С. 5–11.
3. Погребной Ан.В., Погребной В.К. Инвариант графа на основе компактных подграфов и алгоритм его вычисления // Известия Томского политехнического университета. – 2013. – Т. 322. – № 5. – С. 200–204.
4. Погребной В.К. Метод интеграции структурных различий в графовых моделях и его применение для описания структур // Известия Томского политехнического университета. – 2011. – Т. 318. – № 5. – С. 10–16.
5. Погребной В.К. Задача определения оценок сходства структур двух графов на основе выделения общих частей // Известия Томского политехнического университета. – 2013. – Т. 322. – № 5. – С. 194–199.
6. Погребной В.К., Погребной Ан.В. Исследование полиномиальности метода вычисления интегрального описателя структуры графа // Известия Томского политехнического университета. – 2013. – Т. 323. – № 5. – С. 146–151.

Поступила 03.09.2013 г.

UDC 519.171.1

POLYNOMIAL ALGORITHM OF COMPUTING COMPLETE GRAPH INVARIANT ON THE BASIS OF INTEGRAL STRUCTURE DESCRIPTOR

V.K. Pogrebnoy, An.V. Pogrebnoy

Tomsk Polytechnic University

The relevance of the research is caused by the unsolved problem of searching for the complete graph invariant and polynomial algorithm for its computing. The aim of the research is in determining the complete invariant of an ordinary graph on the basis of integral descriptor of abstract structure and in developing the efficient algorithm for computing the complete invariant. The techniques of the research are based on the graph theory and the theory of structural differences code integration in abstract graph structures. The authors have proposed the algorithm for solving one of the most complex problems of graph theory. It is the computation of complete graph invariant. The algorithm is based on the methods of free and dependent integration of structural differences codes in a graph; it is characterized by simplicity, efficiency and it has polynomial estimation of the limiting amount of computation. The complete invariant is represented in the form of a vector of integral descriptor for graph abstract structure vertices and contains information for forming isomorphism substitution. Using Java the GraphISD software was developed implementing the proposed algorithm. The paper introduces the examples of computing the complete invariants at free and dependent integration.

Key words:
Complete graph invariant, graph isomorphism, integral structure descriptor, abstract graph structure, code integration area, polynomial algorithm.

REFERENCES

1. Zykov A.A. *Osnovy teorii grafov* [Graph theory bases]. Moscow, Vuzovskaya kniga, 2004. 664 p.
2. Demer M., Emmert-Shtrayb F., Tsoy U.P., Varmuza K. Noviy funktsional informativnosti dlya analiza struktury khimicheskikh grafov [A new information content functional to analyze the structure of kinetic graphs]. *Bulletin of the Tomsk Polytechnic University*, 2010, vol. 316, no 5, pp. 5–11.
3. Pogrebnoy An.V., Pogrebnoy V.K., Invariant grafa na osnove kompaktnykh podgrafov i algoritm ego vychisleniya [Graph invariant on the basis of compact subgraph and algorithm for its computation]. *Bulletin of the Tomsk Polytechnic University*, 2013, vol. 322, no. 5, pp. 200–204.
4. Pogrebnoy V.K. Metod integratsii strukturnikh razlichii v grafovykh modelyakh i ego primenenie dlya opisaniya struktur [Methods of integration of structural differences in graph models t odescribe structures]. *Bulletin of the Tomsk Polytechnic University*, 2011, vol. 318, no. 5, pp. 10–16.
5. Pogrebnoy V.K. Zadacha opredeleniya otsenok skhodstva struktur dvukh grafov na osnove vydeleniya obshchikh chastei [The task of estimating the similarity of two graphs structures on the basis of general parts definition]. *Bulletin of the Tomsk Polytechnic University*, 2013, vol. 322, no 5, pp. 194–199.
6. Pogrebnoy V.K., Pogrebnoy An.V. Issledovanie polinomialnosti metoda vychisleniya integralnogo opisatelya strukturi grafa [Studying the polynomiality of computation technique for integrated descriptor of graph structure]. *Bulletin of the Tomsk Polytechnic University*, 2013, vol. 323, no. 5, pp. xx–xx.

УДК 004.032.24

РАСПАРАЛЛЕЛИВАНИЕ АЛГОРИТМА ВЫДЕЛЕНИЯ ГРАНИЦ ОБЪЕКТОВ НА ОСНОВЕ СТРУКТУРНО-ГРАФИЧЕСКОГО ПРЕДСТАВЛЕНИЯ

А.Ю. Демин, В.А. Дорофеев

Томский политехнический университет
E-mail: ad@tpu.ru

Актуальность работы *обусловлена необходимостью исследования возможности распараллеливания алгоритмов на основе структурно-графического представления.*
Цель работы*: Распараллелить алгоритм нахождения границ на аэрофотоснимках с помощью структурно-графической формы.*
Методы исследования: *Обработка растровых изображений в интеллектуальных системах навигации и управления с помощью линейных фильтров. Представление алгоритмов в графовой форме: дерева операторов, блок-схемы, графа потока данных. Проектирование программных средств с помощью средств платформы .Net библиотеки Task Parallel Library.*
Результаты: *В работе рассматривается программная реализация алгоритма нахождения границ объектов на изображениях с помощью оператора Собеля. Программная реализация представлена в структурно-графической форме. Предложен полуавтоматический способ распараллеливания рассматриваемой программной нагрузки. Программно реализован распараллеленный алгоритм, и проведен анализ эффективности распараллеливания для различных изображений.*

Ключевые слова:
Оператор Собеля, обработка изображений, дерево операторов, граф потока данных, распараллеливание программ.

Введение

Интеллектуальные системы навигации и управления (ИСНУ) должны обладать современными средствами связи и мониторинга, обеспечивающими сбор, накопление и обработку информации, поступающей с подвижных и стационарных, в том числе труднодоступных, объектов, выработку решений и оперативную доставку команд управления на все уровни и на все объекты, осуществлять автоматизированное документирование информационных обменов, результатов обработки информации и принятых решений, обеспечивать непрерывный мониторинг состояния объектов, отображение местоположения и маршрутов движения мобильных групп и подвижных объектов на электронных картах диспетчерских пунктов в реальном масштабе времени и целый ряд других функций [1].

Использование спутниковых и аэрофотоснимков, изображений, получаемых с беспилотных летательных аппаратов, позволило в телекоммуникационных технологиях существенно улучшить возможности построения аппаратно-программных комплексов для сопровождения, мониторинга и управления мобильными объектами [2].

Важной задачей при анализе изображений, полученных с различных подвижных и труднодоступных объектов, в том числе со спутников, является проблема распознавания различных объектов, таких как различные водоемы (реки, озера, водохранилища, затопленные территории), площади выгоревших лесов, промышленные и гражданские объекты и т. д. Решение подобных задач актуально при построении ИСНУ для Авиалесоохраны РФ, Россельхознадзора РФ, Росгидромета РФ и гидрометов стран СНГ, различных служб МЧС и силовых ведомств.

С другой стороны, изображения, получаемые со спутников и других объектов, характеризуются большим объемом данных. Так, файл с одним