

УДК 519.171.1

## МЕТОД ДИФФЕРЕНЦИАЦИИ ВЕРШИН ГРАФА И РЕШЕНИЕ ПРОБЛЕМЫ ИЗОМОРФИЗМА

**Погребной Андрей Владимирович,**

магистрант кафедры информатики и проектирования систем Института кибернетики Томского политехнического университета, Россия, 634050, г. Томск, пр. Ленина, д. 30. E-mail: avpogrebnoy@gmail.com

**Погребной Владимир Кириллович,**

д-р техн. наук, профессор кафедры информатики и проектирования систем Института кибернетики Томского политехнического университета, Россия, 634050, г. Томск, пр. Ленина, д. 30. E-mail: vkp@tpu.ru

**Актуальность.** Одной из актуальных проблем современной прикладной теории графов является моделирование связей между структурой объекта и его свойствами. Важное место при установлении такой связи занимает проблема идентификации и оценивания сходства структур. Известные подходы к оцениванию сходства структур в энергетике, геоинформационных системах, компьютерной химии, и, в частности в нефтехимии, ограничиваются использованием набора косвенных признаков и не рассматривают возможности решения проблемы, основываясь на прямых признаках, связанных с изоморфизмом.

**Цель научной работы:** сформулировать теоретические основы метода дифференциации вершин графов, показать возможные применения метода для оценки сходства структур и решения проблемы изоморфизма, рассматривая его как частный случай полного сходства структур.

**Методы исследования** основаны на применении прикладной теории графов, теории построения и анализа эффективных алгоритмов, моделирования структур с помощью автоматных моделей, применении теории интеграции кодов структурных различий в процессе дифференциации вершин.

**Результаты.** Сформулирована проблема идентификации структуры графа, объединяющая проблемы инвариантного описания структуры и получения полного инварианта графа, изоморфизма графов, оценивания сходства структур. Показано, что решение перечисленных задач в основном сводится к решению проблемы дифференциации вершин в структуре графа. Введено три вида структурных различий в графе – базовые, скрытые, виртуальные. Предложена автоматная модель структуры графа, положенная в основу метода дифференциации вершин и разработки алгоритмов вычисления полных инвариантов со свободной интеграцией кодов структурных различий (алгоритм ISD-F), зависимой (ISD-D) и независимой (ISD-I). Рассмотрены особенности применения данных алгоритмов при решении проблемы изоморфизма и возможности разработки алгоритма для оценивания сходства структур на основе взаимозависимой интеграции кодов. Проведены экспериментальные исследования алгоритмов при вычислении полных инвариантов и проверке на изоморфизм графов, содержащих до 5000 вершин. Эксперименты показали высокую эффективность алгоритмов при решении этих задач.

### **Ключевые слова:**

Идентификация структуры графа, полный инвариант графа, изоморфизм графов, интегральный описатель структуры, автоматная модель структуры, дифференциация вершин графа.

Проблема установления связей между структурой объекта и его свойствами является одной из важных задач прикладной теории графов. Такие связи широко применимы в компьютерной химии [1–4], энергетике [5], геоинформационных системах [6] и многих других областях. Среди задач моделирования «структура–свойство» важное место занимает проблема идентификации структур. Решение этой проблемы позволит разработать эффективные методы представления, поиска и оценки сходства структур. В данной статье предложен математический инструмент в виде метода дифференциации вершин, составляющий основу решения перечисленных задач, включая получение инвариантного описателя структуры, определение изоморфизма и вычисление полного инварианта.

### **О проблеме идентификации структур графов**

Два обыкновенных графа  $G=(E_G, U_G)$  и  $H=(E_H, U_H)$  с множествами вершин  $E_G, E_H$ ,  $|E_G|=|E_H|$  и рёбер  $U_G, U_H$ ,  $|U_G|=|U_H|$  являются изоморфными, если существует вариант совмещения вершин множеств  $E_G$  и  $E_H$ , при котором все рёбра множеств  $U_G$  и  $U_H$  также

оказываются совмещены. Очевидно, что в изоморфных графах  $G$  и  $H$  совмещаемые вершины должны иметь равные степени. Известно большое число других характеристик, равенство которых в сравниваемых графах является необходимым условием их изоморфизма [7]. На основе учёта таких характеристик разработано большое разнообразие эвристических алгоритмов определения изоморфизма, как правило, ориентированных на отдельные типы графов: для ациклических – это, например [8, 9], для планарных – [10].

В общем случае для определения изоморфизма графов  $G$  и  $H$ , содержащих по  $n$  вершин, необходимо перебрать  $n!$  вариантов совмещения вершин множеств  $E_G$  и  $E_H$ . При таких объёмах вычислений даже весьма значительные успехи эвристических алгоритмов по сокращению этих объёмов для частных случаев в целом остроту проблемы изоморфизма не снижают [11]. Попытки решить задачу тестирования изоморфизма без получения подстановки вершин сравниваемых графов  $G$  и  $H$  также нельзя признать успешными [12]. Так, для двух изоморфных графов Пейли 17-го порядка програм-

ма тестирования, приведённая в [12], показала, что эти графы неизоморфны.

Следует признать, что в теории графов сложилась парадоксальная ситуация, которую можно обозначить как отсутствие возможностей для однозначной идентификации структуры графа. Действительно, принимаясь за исследование очередного графа, мы не можем установить – является ли он одним из ранее рассмотренных. Для однородных графов такая ситуация возникает уже, например, при приближении значений  $n$  к 15–20 вершинам. Что касается неоднородных графов, содержащих десятки и сотни вершин, то для таких размерностей даже применение алгоритмов, учитывающих специфику отдельных графов, объёмы вычислений часто становятся также нереалистичными [11, 13].

Получается, что в теории графов нет инструмента для описания основного объекта исследований – структуры графа. Произвольная нумерация вершин, которая традиционно используется для представления графа в компьютер, не является инвариантным описанием его структуры, т. е. независимым от нумерации вершин. Один и тот же граф с произвольными нумерациями вершин воспринимается как два разных графа.

При анализе структур графов, в частности в компьютерной химии, часто используются числовые характеристики, которые отражают отдельные свойства структур и не зависят от нумерации вершин [14–18]. Такие характеристики названы инвариантами. Примером инварианта является  $n$ -мерный вектор  $S(G)$  с упорядоченными по возрастанию значениями степеней  $s_i$  вершин  $e_i \in E_G$  графа  $G$ . Приведем пример другого инварианта  $M^l(G)$ , вычисляемого на основе матрицы смежности вершин графа  $G$ , возведенной в степень  $l$ . Такую матрицу обозначим  $A^l(G)$ ,  $l=2,3,\dots$ . Элементы  $a_{ij}^l$  матрицы  $A^l(G)$  соответствуют числу маршрутов длины  $l$ , связывающих вершины  $e_i$  и  $e_j$ . Соответственно, инвариант  $M^l(G)$ , полученный на основе матрицы  $A^l(G)$ , будем называть маршрутным. Инвариант  $M^l(G)$ , так же как и степенной инвариант  $S(G)$ , представляется в виде  $n$ -мерного вектора. Элементом вектора  $M^l(G)$  является упорядоченное по возрастанию значений множество ненулевых элементов строк  $A^l(G)$  матрицы  $A^l(G)$ . В векторе  $M^l(G)$  элементы записываются в порядке возрастания значений числовых эквивалентов, полученных для соответствующих множеств.

В теории графов накоплено большое разнообразие инвариантов [7]. Известны попытки получить полный набор инвариантов, идентифицирующих структуру графа с точностью до изоморфизма. Все такие попытки, так же как и исследования по поиску видов полных инвариантов и алгоритмов их вычисления, до последнего времени оказывались безуспешными [19, 20]. Известен только один полный инвариант в виде миникода [21], но для его вычисления, как и для определения изоморфизма двух графов  $G$  и  $H$ , требуется перебор  $n!$  вариантов перестановок строк и столбцов матрицы  $A(G)$ .

Решение такой сложной задачи, какой, несомненно, является проблема получения полного инварианта графа, как это часто и случается в научной работе, оказалось неожиданно простым и оригинальным [22–24]. В работе [24] предлагается алгоритм вычисления полных инвариантов  $P(G)$  для однородных и неоднородных обыкновенных графов в виде интегрального описателя структуры (Integral Structure Descriptor – ISD). В этом случае полный инвариант  $P(G)$  выступает в роли однозначного идентификатора абстрактной структуры графа  $G$ . Равенство идентификаторов  $P(G)$  и  $P(H)$  графов  $G$  и  $H$  означает, что они изоморфны, т. е. их абстрактные структуры совпадают.

Экспериментальные исследования алгоритма вычисления  $P(G)$  в [24] выполнялись с помощью специально разработанной программы GraphISD на графах, содержащих до 1000 вершин. Результаты экспериментов, приведенных в [24], подтверждают высокую эффективность работы алгоритма. Например, для однородного графа  $G$  с числом вершин  $n=1000$  и степенью  $s=200$ , время вычисления  $P(G)$  без привлечения маршрутной характеристики (самый трудный случай) составило 63564 мс, а с применением маршрутной характеристики – 274 мс. Для неоднородных графов алгоритм работает существенно быстрее. В частности, для  $n=2000$  с множеством значений степеней (100,200), генерируемых вершинам примерно в равных долях, потребовалось 410 мс, а для  $n=10000$  с множеством значений степеней (100,200,300,400) – 4900 мс. Приведенные выше значения времени вычисления полного инварианта даны без учёта времени на генерацию графов.

Наличие эффективного алгоритма вычисления полного инварианта автоматически решает проблему изоморфизма графов. Этому, несомненно, можно порадоваться, но задачи анализа структур графов далеко не ограничиваются проверкой их на изоморфизм. В частности, проблема сходства структур графов для практических применений является более важной и обобщает задачу определения изоморфизма, рассматривая её как частный случай, который соответствует полному сходству графов.

Данная статья представляет первые результаты исследований по проблеме сходства структур графов с равными степенными инвариантами. В основу исследований положен метод дифференциации вершин графа. Основное внимание уделено проверке наличия полного сходства структур сравниваемых графов, что соответствует их изоморфизму. Идея метода дифференциации вершин с помощью рекуррентного правила интеграции кодов структурных различий была положена в основу разработки алгоритмов вычисления полных инвариантов [22–24]. В этих работах авторами было подмечено, что данная идея может оказаться весьма полезной при разработке методов анализа сходства структур графов. Поэтому в статье, отражающей начальную стадию исследований, основное

внимание было уделено установлению сходства процессов дифференциации вершин сравниваемых графов.

Если сходство процессов наблюдается до полной дифференциации вершин, то сравниваемые графы изоморфны. Отсутствие сходства на любом шаге процесса означает, что графы не изоморфны. Такой подход, в отличие от установления изоморфизма графов  $G_1$  и  $G_2$  на основе сравнения их полных инвариантов  $P(G_1)$  и  $P(G_2)$ , позволяет обнаружить отсутствие изоморфизма непосредственно в процессе дифференциации вершин и, при необходимости, перейти к анализу сходства структур в сравниваемых графах.

В статье предлагается представлять структуру графа в виде динамической системы так, чтобы её функционирование воспроизводило процесс дифференциации вершин. При работе системы модель структуры сама о себе собирает информацию. В итоге каждая вершина графа получает уникальный код (дескриптор). Коды интегрируют структурные характеристики (различия) и особенности расположения каждой вершины относительно всех других вершин в абстрактной структуре графа.

Статья содержит определение понятия вычисляемого, скрытого и виртуального структурного различия. Рассматриваются правила их применения при дифференциации вершин. Подробно изложена динамика процесса дифференциации на основе интеграции кодов структурных различий и его алгоритмизация в вариантах свободной и зависимой интеграции. Приводятся также результаты экспериментов при определении изоморфизма графа на основе анализа сходства структур.

### Структурные различия и дифференциация вершин графа

Обыкновенные графы соответствуют базовому понятию структуры, как совокупности вершин и бинарных отношений между ними. При этом вершины и отношения в структуре не помечены какими-либо атрибутами, что часто используется в графовых моделях для практических приложений. Решая проблему идентификации структур для обыкновенных графов, мы рассматриваем самый трудный случай. Относительно других видов графов (ориентированные, двудольные, ациклические и т. д.), в том числе и при наличии атрибутов, решение данной проблемы существенно упрощается.

Идентификатор структуры обыкновенного графа в виде полного инварианта легко получается, если удастся подобрать одну или несколько структурных характеристик, которые приводят к полной дифференциации вершин. На рис. 1 приведен пример графа  $H$ , для которого показана дифференциация вершин на основе степенной  $\tilde{S}(H)$  и маршрутной  $\tilde{M}^2(H)$  характеристик.

По характеристике  $\tilde{S}(H)$  вершины графа  $H$  разбиваются на 6 групп:  $(e_1, e_8)$ ,  $(e_2, e_4)$ ,  $(e_7)$ ,  $(e_5)$ ,  $(e_6)$ ,  $(e_3)$ . Характеристика  $\tilde{M}^2(H)$  полностью дифференцирует вершины графа. Если в множествах  $\tilde{S}(H)$  и  $\tilde{M}^2(H)$  элементы упорядочить по возрастанию значений, то получим степенной инвариант  $S(H)$  и маршрутный  $M^2(H)$ . Порядковые номера элементов в инварианте на рис. 1 указаны справа от характеристики  $\tilde{M}^2(H)$  в виде вектора  $D^0$ . Для однородного графа  $G$ , представленного на рис. 1, маршрутная характеристика  $\tilde{M}^2(G)$  разбивает вершины всего лишь на 3 группы:  $(e_3, e_7)$ ,  $(e_1, e_2, e_6, e_8)$ ,  $(e_4, e_5)$ .

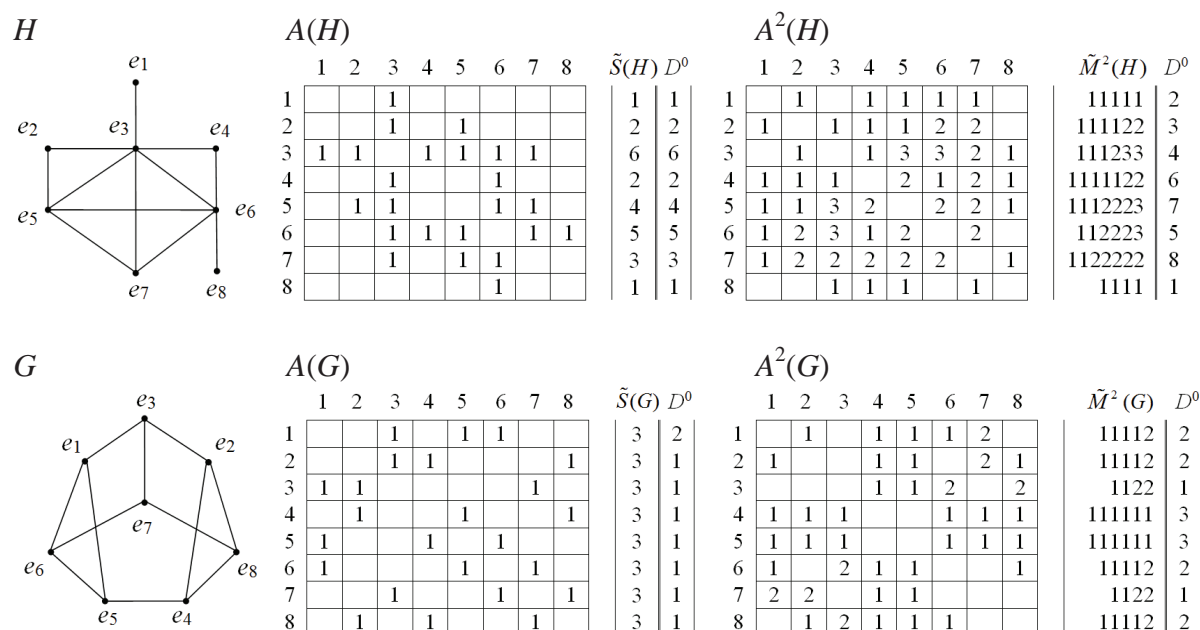


Рис. 1. Примеры дифференциации вершин на основе степенных и маршрутных характеристик

Fig. 1. Examples of vertices differentiation based on power and routing characteristics

Вектор  $D^0$  графа  $H$ , полученный в условиях полной дифференциации вершин на основе характеристики  $M^2(H)$ , принимается, в соответствии с алгоритмом [18], в качестве вектора  $D(H)$ , отражающего финальную стадию процесса дифференциации. Если граф  $H$  на рис. 1 представить списком инцидентов  $F(e_j)$  вершин  $e_j \in E_H$  в виде  $\{e_j(F(e_j))\}$ , то, подставляя вместо  $e_j$  соответствующие значения кодов  $d_j$  вектора  $D(H)$ , получим список  $\tilde{P}(H) = \{d_j(F(d_j))\}$ . После упорядочения в списке  $\tilde{P}(H)$  элементов  $d_j(F(d_j))$  по возрастанию значений  $d_j$  получим полный инвариант  $P(H) = \{d_j(F(d_j))\}$ . Для нашего примера полный инвариант  $P(H) = \{1(5), 2(4), 3(4,7), 4(2,3,5,6,7,8), 5(1,4,6,7,8), 6(4,5), 7(3,4,5,8), 8(4,5,7)\}$ . Инвариант  $P(H)$  является идентификатором или инвариантным описателем абстрактной структуры графа  $H$  с номерами вершин  $d_j$  и рёбрами, определяемыми инцидентами  $F(d_j)$  вершин  $d_j$ .

Из приведенного выше примера следует, что в деле получения полного инварианта главным является достижение полной дифференциации вершин. Для решения данной задачи необходимо найти основание для различия вершин на структурном уровне. Характеристики вершин и отношений между ними в структуре графа, которые способны привести к частичной или полной дифференциации вершин будем именовать структурными различиями. Из примеров на рис. 1 следует, что значения элементов степенной и маршрутных характеристик соответствуют определению структурного различия. Степенная характеристика частично дифференцирует вершины неоднородного графа (граф  $H$  на рис. 1), а маршрутная может привести к дифференциации вершин однородного графа (граф  $G$  на рис. 1).

Для дифференциации вершин могут быть использованы и другие известные в теории графов вычисляемые характеристики. Структурные различия вершин на основе вычисляемых характеристик будем именовать базовыми. Для достижения большей дифференциации можно одновременно использовать совокупность базовых различий. Но в этом случае нельзя гарантировать достижение полной дифференциации. Чтобы продолжить дифференциацию вершин, очевидно, следует учитывать их различия в отношениях. Описать все многообразие структурных различий в отношениях, определяемых особенностями структур графов, не представляется возможным. Поэтому структурные различия такого вида будем именовать скрытыми.

Мы знаем, что скрытые различия существуют и могут быть использованы для дифференциации вершин. Например, вершины  $e_2$  и  $e_4$  в графе  $H$  на рис. 1 неразличимы по степенной характеристике  $\tilde{S}(H)$ , но  $e_2$  связана с  $e_3$ , у которой  $s_3=4$ , а у  $e_4$  нет связи с вершиной со степенью 4. Проблема заключается в том, что для идентификации графов на основе полной дифференциации необходимо все скрытые различия описать и использовать наравне

с базовыми различиями. Решение данной проблемы будет рассмотрено в следующем разделе статьи.

Применение базовых, а затем и скрытых различий также не гарантирует достижение полной дифференциации вершин. Например, в симметричных графах не могут работать оба вида различий. Частичная дифференциация вершин, полученная после применения базовых и скрытых структурных различий, означает, что ряд вершин графа по отношению к данным базовым различиям и всем скрытым различиям, порождаемым структурой графа, оказались неразличимыми. Для дифференциации неразличимых вершин предлагается вводить дополнительное искусственное различие, которое будем именовать виртуальным.

Каждое из трех видов различий в дифференциации вершин играет свою роль. Базовые и виртуальные различия применяются для получения начальной частичной дифференциации, а скрытые – для её продолжения. Основную проблему представляют скрытые различия – отсутствие возможности их выделения и применения для дифференциации. Ниже предложен метод выполнения процесса дифференциации, в котором структура графа рассматривается как некоторая динамическая система, функционирующая в дискретном времени. В ходе функционирования такой системы интегрируется и накапливается информация о скрытых различиях относительно положения вершин в структуре графа.

#### Процесс дифференциации вершин на основе модели в виде динамической системы

Функционирование структуры графа как динамической системы, порождающее дифференциацию вершин на основе скрытых различий, может осуществляться только при наличии начальной частичной дифференциации. Для получения начальной дифференциации используется априорно заданная совокупность базовых различий. Такие различия должны легко вычисляться. Например, это базовые различия на основе степенных  $\tilde{S}(H)$  или маршрутных  $M^2(H)$  характеристик. Если с помощью этих базовых различий получить начальную дифференциацию не удается, то вводится виртуальное различие.

Начальная дифференциация вершин принимается как исходное состояние динамической системы и представляется в виде вектора  $D^0 = \{d_i^0\}$ ,  $i=1,2,\dots,n$ ,  $n$  – число вершин в графе. Элементы  $d_i^0$  обозначают порядковые номера упорядоченных по возрастанию значений характеристики, используемой в качестве базового различия. Например, для графа  $G$  на рис. 1, по характеристике  $M^2(G)$ , вектор  $D^0$  отражает три различия,  $d_i^0=1,2,3$ , а по характеристике  $\tilde{S}(G)$  нет различий, и для получения вектора  $D^0$  в граф  $G$  через вершину  $e_1$  вводится виртуальное различие  $d_1^0=2$ . В графе  $H$  на рис. 1 по характеристике  $\tilde{S}(H)$  вектор  $D^0$  отражает шесть различий. При этом элементы  $d_i^0$  вектора



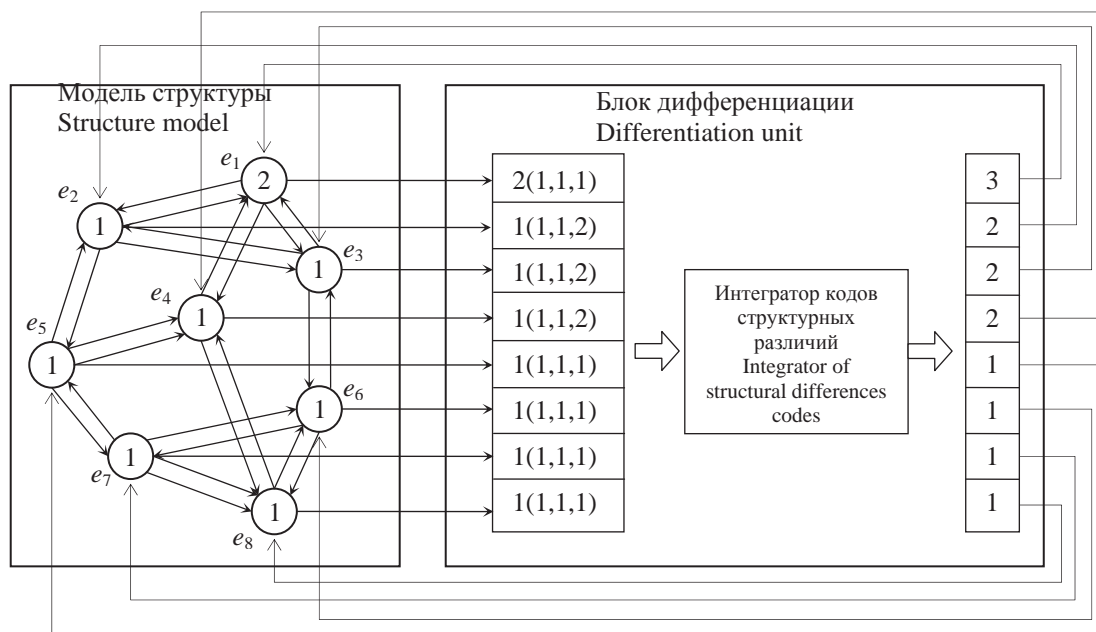


Рис. 2. Динамическая система структуры графа  $G$

Fig. 2. Dynamic system of  $G$  graph structure

$D^0$  оказались равными значениям степеней  $s_i$  вершин  $e_i$ . По характеристике  $\tilde{M}^2(H)$  достигнута полная дифференциация, т. е. вектор  $D^0$  принимается в качестве конечного вектора  $D$  и запускать процесс дифференциации на динамической системе структуры графа  $H$  нет необходимости.

В динамической системе структуры графа каждая вершина соответствует автомату. Автоматы в системе взаимодействуют между собой по каналам связи, которые соответствуют рёбрам графа. Для перехода системы из одного состояния в другое все автоматы выполняют единый набор действий под управлением блока дифференциации. Пример динамической системы для дифференциации вершин графа  $G$  с вектором  $D^0$  на основе виртуального различия приведен на рис. 2.

Динамическая система состоит из 2-х частей: автоматная модель структуры графа и блок управления процессом дифференциации. Автомат вершины  $e_i$  при переходе системы из состояния  $D^k$  в  $D^{k+1}$  последовательно выполняет следующий цикл действий:

- приём от блока управления дифференциацией состояния  $d_i^k$ ;
- передача состояния  $d_i^k$  всем автоматам вершин  $e_j$ , связанным с автоматом вершины  $e_i$ ;
- приём от автоматов вершин  $e_j$  состояний  $d_j^k$  и формирование из них упорядоченного по возрастанию значений  $d_j^k$  множества  $F(d_i^k)$  – инцидентора вершины  $e_i$  с состоянием  $d_i^k$ ;
- передача в блок управления дифференциацией инцидентора  $F(d_i^k)$  и состояния  $d_i^k$  в форме  $d_i^k(F(d_i^k))$ .

Блок управления дифференциацией организует работу динамической системы и на основе дан-

ных, полученных в результате выполнения очередного цикла действий автоматной модели, вычисляет новое состояние системы. При этом выполняются следующие операции:

- передача состояний  $\{d_i^k\}$  автоматам вершин модели (при запуске процесса дифференциации автоматам передается начальное состояние  $D^0 = \{d_i^0\}$ );
- приём данных от автоматов модели в форме  $\{d_i^k(F(d_i^k))\}$ ;
- вычисление нового состояния  $D^{k+1}$  на основе собранных данных  $\{d_i^k(F(d_i^k))\}$  и ряда параметров, определяющих разные условия процесса дифференциации.

Функционирование динамической системы можно представить последовательностью смены состояний  $D^0, D^1, \dots, D^k, \dots, D^{k^*} = D$ ,  $k^* < n$ , от начального  $D^0$  до конечного  $D^{k^*} = D$ , соответствующего полной дифференциации вершин. При запуске процесса дифференциации система переводится в состояние  $d_i^0$ . Далее автоматы обмениваются между собой состояниями  $d_i^0$  и формируют инциденторы  $F(d_i^0)$ , которые в форме  $d_i^0(F(d_i^0))$  передаются в блок управления. В этом блоке на основе анализа данных  $\{d_i^0(F(d_i^0))\}$  определяется новое состояние системы  $D^1$ . После этого система переводится в новое состояние  $D^1$  вместо  $D^0$  и цикл действий по формированию данных  $\{d_i^1(F(d_i^1))\}$  повторяется. Таким образом, при функционировании динамической системы циклически происходит накопление и интеграция данных о структурных различиях и на этой основе дифференциация вершин.

В общем случае смена состояния системы  $D^k \Rightarrow D^{k+1}$  осуществляется с помощью алгоритма оператора  $A_z$ , т. е.  $D^{k+1} = A_z\{d_i^k(F(d_i^k))\}$ . Индекс  $z$  ука-

зывает на модификацию алгоритма в зависимости от задачи, в составе которой выполняется процесс дифференциации вершин. Задача базового алгоритма  $A$  заключается в получении интегрального описателя структуры (ISD). Модификации алгоритма учитывают особенности задач, что приводит к необходимости введения дополнительных операций и изменениям правила выполнения процесса дифференциации вершин. К настоящему времени выделено 4 задачи, каждая из которых использует алгоритм получения ISD в соответствующей модификации.

1. Получение для графа  $G$  полного инварианта в условиях «свободного» назначения кодов (дескрипторов) инциденторам  $d_i^k(F(d_i^k))$ , отражающим структурные различия. В этом случае процесс дифференциации сопровождается запоминанием принятых кодов и оформлением их в систему кодирования  $W_G(S)$ . Здесь  $S$  обозначает степенной инвариант графа  $G$ , а индекс  $G$  указывает на принадлежность системы кодирования графу  $G$ . Полный инвариант графа  $G$ , полученный в условиях свободной интеграции кодов структурных различий, будем обозначать  $P_w(G)$ , подчеркивая зависимость такого инварианта от системы кодирования  $W_G(S)$ . Алгоритм вычисления полного инварианта  $P_w(G)$ , использующего свободную (free) интеграцию кодов, будем обозначать ISD-F.
2. Поиск в графе  $H$  полного инварианта  $P_G(H)=P_w(G)$  с использованием в процессе дифференциации системы кодирования  $W_G(S)$ . Индекс  $G$  в обозначении  $P_G(H)$  указывает на применение в процессе интеграции кодов системы кодирования  $W_G(S)$ , полученной для графа  $G$ . Наличие инварианта  $P_G(H)=P_w(G)$  означает, что граф  $H$  изоморфен графу  $G$ . Если инвариант  $P_G(H)=P_w(G)$  найти не удалось, то на основе анализа процесса интеграции и достигнутой дифференциации вершин могут быть получены некоторые оценки сходства графа  $G$  и  $H$ . При необходимости для графа  $H$  может быть получен полный инвариант  $P_w(H)$  со своей системой кодирования  $W_H(S)$ . При поиске инварианта  $P_G(H)$  применялась зависимая (dependent) от  $W_G(S)$  интеграция кодов структурных различий. В соответствии с этим алгоритм поиска инварианта  $P_G(H)$  назван ISD-D.
3. Вычисление для графа  $G$  полного инварианта  $P(G)$  без формирования или использования какой-либо системы кодирования. В этой задаче алгоритм при дифференциации вершин использует независимую (independent) интеграцию кодов структурных различий и получил название ISD-I. Равенство полных инвариантов  $P(G)$  и  $P(H)$  является гарантией изоморфизма графов  $G$  и  $H$ . Пользуясь алгоритмом ISD-I, мы получаем для любого графа  $G$  полный инвариант  $P(G)$ , который одновременно является инвариантным описателем абстрактной структуры данного графа.

4. Решение проблемы оценивания сходства абстрактных структур графов с применением метода дифференциации вершин. В этой проблеме изоморфизм двух графов соответствует полному сходству их структур и рассматривается как частный результат, полученный при оценивании сходства. Анализ сходства структур предлагается осуществлять с помощью метода дифференциации вершин, в котором процесс интеграции кодов структурных различий в сравниваемых структурах графов производится одновременно. При этом оценивание сходства делается постоянно на основе сопоставления процессов дифференциации после очередной смены состояний динамических систем структур графов. Процессы интеграции в этом случае являются взаимозависимыми (interdependent) от структур сравниваемых графов, а соответствующая модификация алгоритма получила название ISD-ID.

Исследования по проблеме сходства структур с помощью метода дифференциации вершин на основе взаимозависимой интеграции кодов структурных различий еще предстоит выполнить. Предлагаемое в статье решение проблемы изоморфизма графов на основе дифференциации вершин, которое будет изложено в следующем разделе, следует рассматривать как начальную стадию этих исследований. Решение здесь базируется на проверке сходства процессов свободной и зависимой интеграции (задачи 1 и 2), что также соответствует целям взаимозависимой интеграции. Что касается алгоритма ISD-I вычисления полного инварианта на основе независимой интеграции (задача 3), то он подробно представлен в работе [24].

#### Решение проблемы изоморфизма графов

Изоморфизм двух обыкновенных графов  $G$  и  $H$  с равными степенными инвариантами  $S(G)=S(H)$  легко установить, проверяя соблюдение равенства  $P_w(G)=P_w(H)$  или  $P(G)=P(H)$ . В первом случае сравнивается полный инвариант  $P_w(G)$ , полученный с помощью алгоритма ISD-F, и полный инвариант  $P_w(H)$ , поиск которого осуществляется с помощью алгоритма ISD-D относительно системы кодирования  $W_G(S)$ . Во втором случае полные инварианты  $P(G)$  и  $P(H)$  вычисляются независимо друг от друга с помощью алгоритма ISD-I.

В алгоритме ISD-I, в отличие от ISD-F и ISD-D, нет необходимости сопоставления процессов дифференциации вершин графов  $G$  и  $H$  на предмет оценивания их сходства, т. к. здесь нет общей системы кодирования. Не исключено, что в последующем будут найдены основания и правила сопоставления процессов дифференциации и для данного алгоритма. Что касается алгоритмов ISD-F и ISD-D, то здесь процесс дифференциации выполняется относительно общей системы кодирования и для алгоритма ISD-D операции сопоставления состояний динамической системы являются обязательными. Поэтому на стадии поиска подходов к решению

проблемы сходства структур большой интерес представляют операции проверки сходства процессов дифференциации на основе свободной и зависимой интеграции, выполняемые алгоритмами ISD-F и ISD-D для установления изоморфизма графов. Заметим также, что по показателям эффективности применение данных алгоритмов для решения проблемы изоморфизма обладает рядом преимуществ в сравнении с алгоритмом ISD-I. Эти преимущества будут отмечены ниже.

По содержанию алгоритм ISD-F полностью воспроизводит работу динамической системы, представленной на рис. 2. Начальное состояние  $D^0$  определяется на основе степенной характеристики  $\hat{S}(G)$  для неоднородных графов и маршрутной характеристики  $\hat{M}^c(G)$  для однородных. Если начальная дифференциация при этом не достигается, то через вершину  $e_1$  вводится виртуальное различие  $d_1^0=2$ .

Рассмотрим правила вычисления нового состояния  $D^{k+1}$  на основе данных  $\{d_i^k(F(d_i^k))\}$ . В состоянии  $D^k$  среди кодов  $d_i^k$ , для которых еще не присвоен новый код выбираются коды  $d_{i^*}^k$  с минимальным значением  $d$ . Группу вершин  $e_{i^*}$  с выбранным кодом  $d$  обозначим  $E_d^k$ . Если  $|E_d^k|=1$ , то  $d_i^{k+1}=d_i^k$  и выбор следующих по величине кодов повторяется. При  $|E_d^k|>1$  инциденты  $F(d_{i^*}^k)$  для  $e_{i^*} \in E_d^k$  преобразуются в числовые эквиваленты и упорядочиваются по возрастанию значений. Вершине  $e_{i^*}$ , соответствующей минимальному из них, присваивается код  $d_i^{k+1}=d$ . Для всех  $e_{i^*}$  следующего по величине числового эквивалента – код  $d_i^{k+1}=d+1$ , затем  $d+2$  и до  $d+\delta$ . Равным числовым эквивалентам присваиваются равные коды. Далее в состоянии  $D^k$  значения всех  $d_i^k > d$  увеличиваются на  $\delta$ , и процесс выбора  $d_{i^*}^k$  с минимальным значением  $d$  повторяется.

После получения нового состояния  $D^{k+1}$  алгоритм ISD-F запоминает соответствия между  $d_i^k(F(d_i^k))$  и присвоенными значениями  $d_i^{k+1}$  и заносит их в систему кодирования  $W_G(S)$ . При достижении состояния  $D$  алгоритм ISD-F формирует полный инвариант в виде  $P_W(G)=\{d_i(F(d_i))\}$ .

В ходе продвижения системы от состояния  $D^0$  к  $D^{k^*}=D$  может возникнуть ситуация, когда  $D^{k+1}=D^k$ , т. е. очередная смена состояний не приводит к дополнительной дифференциации вершин. В этом случае состояние  $D^k$  называется устойчивым и

обозначается как  $D^{k^*}$ . Здесь индекс  $v=1,2,\dots,V$  указывает на порядковый номер устойчивого состояния, возникшего на пути от  $D^0$  к  $D$ . Группы вершин  $E_d^{k^*}$  с равными значениями  $d_i^{k^*}$  называются устойчивыми. Такие группы играют важную роль в теории метода ISD и достаточно подробно рассмотрены в работе [23]. Для продолжения процесса дифференциации в состоянии  $D^{k^*}$  выбирается устойчивая группа  $E_d^{k^*}$  минимальной мощности и с минимальным значением  $d$ . Через одну из вершин этой группы вводится виртуальное различие с кодом, равным  $\max\{d_i^{k^*}\}+1$ . После этого процесс дифференциации продолжается до следующего устойчивого состояния  $D^{k^*+1}$  либо до  $D^{k^*}=D$ .

Результат работы алгоритма ISD-F для графа  $G$ , приятного в качестве модели на рис. 2, представлен на рис. 3. Для достижения полной дифференциации потребовалось три смены состояний. При этом все состояния оказались без устойчивых групп. Если виртуальное различие ввести через вершину  $e_5$ , то состояние  $D^3$  будет содержать три устойчивые группы по две вершины. Система кодирования  $W_G(S)$  на рис. 3 не показана. Она содержит все соответствия между  $d_i^k(F(d_i^k))$  и  $d_i^{k+1}$ . Например,  $d_4^1(F(d_4^1))=2(1,1,3)$  соответствует код  $d_4^2=4$ .

Полный инвариант  $P_W(G)=\{d_i(F(d_i))\}$ , полученный для графа  $G$  на основе состояния  $D(G)$ , сопровождается системой кодирования  $W_G(S)$  и выступает в роли эталона при проверке изоморфизма графа  $H$  графу  $G$ . Графы  $H$  и  $G$  являются изоморфными, если между процессами дифференциации вершин графа  $H$  и графа  $G$  имеет место полное сходство. В этом случае полный инвариант  $P_G(H)=P_W(G)$ . Поиск для графа  $H$  полного инварианта  $P_G(H)=P_W(G)$  осуществляется с помощью алгоритма ISD-D. Рассмотрим отличия алгоритма ISD-D от ISD-F.

Первое отличие связано с изменением правила вычисления нового состояния  $D^{k+1}$ . Будем считать, что состояние  $D^k(H)=\{d_i^k\}$  для графа  $H$  равно состоянию  $D^k(G)=\{d_i^k\}$  для графа  $G$ , если между кодами  $d_i^k$  и  $d_i^k$  можно установить взаимно однозначное соответствие, при котором  $d_j^k=d_i^k$ . Проверку наличия такого соответствия при каждой смене состояния в процессе дифференциации вершин графа  $H$  выполняет алгоритм ISD-D. Для этого после полу-

$G = \{e_i(F(e_i))\}$	$d_i^0(F(d_i^0))$	$d_i^1(F(d_i^1))$	$d_i^2(F(d_i^2))$	$d_i^3(F(d_i^3))$	$P_W(G)$
1(2,3,4)	2(1,1,1)	3(2,2,2)	6(4,5,5)	8(5,6,7)	1(2,3,4)
2(1,3,5)	1(1,1,2)	2(1,2,3)	5(3,5,6)	7(4,6,8)	2(1,3,5)
3(1,2,6)	1(1,1,2)	2(1,2,3)	5(2,5,6)	6(3,7,8)	3(1,2,6)
4(1,5,8)	1(1,1,2)	2(1,1,3)	4(2,3,6)	5(2,4,8)	4(1,5,7)
5(2,4,7)	1(1,1,1)	1(1,2,2)	3(1,4,5)	4(1,5,7)	5(2,4,8)
6(3,7,8)	1(1,1,1)	1(1,1,2)	2(1,2,5)	3(1,2,6)	6(3,7,8)
7(5,6,8)	1(1,1,1)	1(1,1,1)	1(2,2,3)	1(2,3,4)	7(4,6,8)
8(4,6,7)	1(1,1,1)	1(1,1,2)	2(1,2,4)	2(1,3,5)	8(5,6,7)

Рис. 3. Пример работы алгоритма ISD-F

Fig. 3. Example of ISD-F algorithm operation

чения данных  $\{d_j^k(F(d_j^k))\}$  каждому элементу  $d_j^k(F(d_j^k))$  ставится в соответствие элемент  $d_i^k(F(d_i^k))$  из системы кодирования  $W_G(S)$  и код  $d_i^{k+1}$  этого элемента присваивается коду  $d_j^{k+1}$ . Равенство состояний  $D^{k+1}(H)$  и  $D^{k+1}(G)$  является основанием для продолжения процесса дифференциации. Если равенства нет, то анализируется две ситуации. Одна из них соответствует завершению процесса поиска инварианта  $P_G(H)=P_W(G)$ . Это означает, что графы  $H$  и  $G$  неизоморфны. Вторая ситуация предусматривает продолжение поиска инварианта путем запуска нового процесса дифференциации.

Необходимость запуска нового процесса дифференциации связана с введением виртуального различия и обуславливает второе отличие алгоритма ISD-D. В алгоритме ISD-F виртуальное различие вводится в устойчивую группу  $E_d^k(G)$  через произвольно выбранную вершину. Из свойств устойчивых групп известно [23], что в подстановке изоморфизма графов  $G$  и  $H$  вершины группы  $E_d^k(G)$  будут соответствовать вершинам группы  $E_d^k(H)$ . При этом неизвестно какая из вершин  $e_j$  в группе  $E_d^k(H)$  будет соответствовать выбранной в  $E_d^k(G)$  вершине  $e_i$ . Поэтому при поиске инварианта  $P_G(H)=P_W(G)$  приходится последовательно вводить виртуальное различие через все вершины  $e_j$  устойчивой группы  $E_d^k(H)$ .

Алгоритм ISD-D организует последовательный выбор вершин устойчивой группы и запуск процесса дифференциации с новым состоянием  $D^k$ . Для всех достигнутых при этом состояний  $D^{k+1}$  и равных соответствующему состоянию в системе  $W_G(S)$  процессы дифференциации запускаются по-

следовательно относительно каждой вершины устойчивой группы, выбранной в состоянии  $D^{k+1}$ . В условиях, когда графы  $G$  и  $H$  являются однородными, в графе  $H$  приходится запускать процессы дифференциации после введения виртуального различия последовательно через все вершины пока не будет найден инвариант  $P_G(H)=P_W(G)$ , либо будет установлено, что графы  $G$  и  $H$  неизоморфны.

На рис. 4 приведен результат работы алгоритма ISD-D по установлению изоморфизма графа  $H$  графу  $G$ , для которого ранее был получен полный инвариант  $P_W(G)$  (рис. 3). В качестве графа  $H$  взят граф  $G$  с измененной нумерацией вершин, т. е. граф  $H$  должен содержать полный инвариант  $P_G(H)=P_W(G)$ . Из рис. 4 следует, что при введении виртуального различия через вершину  $e_1$  уже в состоянии  $D^1$  данные  $\{d_i^1(F(d_i^1))\}$  графа  $H$  не соответствуют данным  $\{d_i^1(F(d_i^1))\}$  графа  $G$ . Второй запуск процесса дифференциации после введения виртуального различия через вершину  $e_2$  привел к получению полного инварианта  $P_G(H)=P_W(G)$ . Абстрактная структура графов  $G$  и  $H$ , построенная по их инвариантам, приведена на рис. 4. Коды  $d_i$ , указанные внутри вершин абстрактной структуры, отражают состояние полной дифференциации. Рядом с вершинами приведены пары  $(e_i, e_j)$ , соответствующие подстановке изоморфизма графов  $G$  и  $H$ . Пары  $(e_i, e_j)$  устанавливаются на основе сопоставления данных  $\{d_i^3(F(d_i^3))\}$  (рис. 3) и  $\{d_j^3(F(d_j^3))\}$  (рис. 4).

Решение проблемы изоморфизма с помощью алгоритмов ISD-F и ISD-D по сравнению с алгоритмом ISD-I обладает рядом преимуществ:

$H=\{e_j(F(e_j))\}$	$d_j^0(F(d_j^0))$	$d_j^1(F(d_j^1))$	$d_j^2$
1(2,7,8)	2(1,1,1)	3(2,2,2)	6
2(1,3,4)	1(1,1,2)	2(1,1,3)	4
3(2,4,8)	1(1,1,1)	2(1,2,2)	3
4(2,3,5)	1(1,1,1)	2(1,1,2)	2
5(4,6,7)	1(1,1,1)	2(1,1,2)	2
6(5,7,8)	1(1,1,1)	2(1,2,2)	?
7(1,5,6)	1(1,1,2)	2(1,1,3)	?
8(1,3,6)	1(1,1,2)	2(1,1,3)	?

$H=\{e_j(F(e_j))\}$	$d_j^0(F(d_j^0))$	$d_j^1(F(d_j^1))$	$d_j^2(F(d_j^2))$	$d_j^3(F(d_j^3))$	$P_G(H)$	$(e_i, e_j)$
1(2,7,8)	1(1,1,2)	2(1,1,3)	4(2,3,6)	5(2,4,8)	1(2,3,4)	(7, 6)
2(1,3,4)	2(1,1,1)	3(2,2,2)	6(4,5,5)	8(5,6,7)	2(1,3,5)	(8, 7)
3(2,4,8)	1(1,1,2)	2(1,2,3)	5(3,5,6)	7(4,6,8)	3(1,2,6)	(6, 5)
4(2,3,5)	1(1,1,2)	2(1,2,3)	5(2,5,6)	6(3,7,8)	4(1,5,7)	(5, 8)
5(4,6,7)	1(1,1,1)	1(1,1,2)	2(1,2,5)	3(1,2,6)	5(2,4,8)	(4, 1)
6(5,7,8)	1(1,1,1)	1(1,1,1)	1(2,2,3)	1(2,3,4)	6(3,7,8)	(3, 4)
7(1,5,6)	1(1,1,1)	1(1,1,2)	2(1,2,4)	2(1,3,5)	7(4,6,8)	(2, 3)
8(1,3,6)	1(1,1,1)	1(1,2,2)	3(1,4,5)	4(1,5,7)	8(5,6,7)	(1, 2)

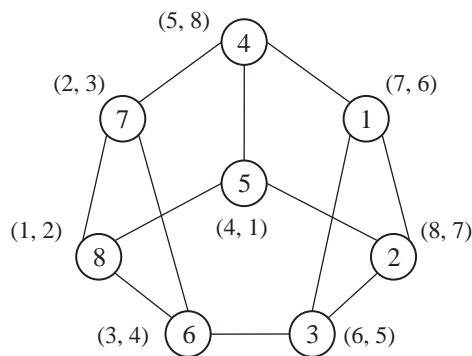


Рис. 4. Пример работы алгоритма ISD-D

Fig. 4. Example of ISD-D algorithm operation



1. Объём вычислений, требуемый для получения инварианта  $P_w(G)$ , существенно меньше, чем для получения инварианта  $P(G)$ . Это особенно ощутимо для однородных графов. В этом случае алгоритму ISD-I приходится делать попытку получить начальную дифференциацию с помощью маршрутной характеристики  $M^2(G)$  или вводить виртуальное различие, при котором процесс дифференциации запускается  $n$  раз, а в алгоритме ISD-F – один раз.
2. Процесс проверки изоморфизма графов  $G$  и  $H$  заканчивается, как только алгоритму ISD-D в графе  $H$  удаётся найти инвариант  $P_G(H)=P_w(G)$ . По объёму вычислений алгоритм ISD-D сравним с алгоритмом ISD-I лишь в условиях, когда требуется выполнять поиск в графе  $H$  всех других инвариантов  $P_G(H)=P_w(G)$ .
3. Основным преимуществом, о котором уже упоминалось ранее, является возможность сопоставлять процессы дифференциации и оценивать сходство структур неизоморфных графов. Проверка изоморфных графов здесь рассматривается как частный результат оценивания сходства структур. Разработка методов оценивания на основе сопоставления процессов дифференциации вершин является предметом отдельных исследований. Некоторые соображения по этой проблеме были приведены при характеристике 4-й задачи в предыдущем разделе. Решение проблемы оценивания сходства структур авторы связывают с дальнейшим развитием концепции сопоставления процессов дифференциации и разработкой методов взаимозависимой интеграции кодов структурных различий под конкретные задачи оценивания сходства структур.

### Результаты экспериментов

Экспериментальные исследования алгоритмов проводились с помощью программы GraphISD, написанной на языке C++, на компьютере, со следующими характеристиками: Intel Core i7–4770 3,40 GHz, 16 GB RAM. Программа включает возможность генерации графов  $G$  с заданными степенными инвариантами  $S(G)$  и перенумерации у них вершин для получения графов  $G^*$  изоморфных графам  $G$ . Начальное состояние  $D^0$  для неоднородного графа формируется на основе степенной характеристики  $\check{S}(G)$ . Для однородного графа – на основе маршрутной характеристики  $\check{M}^2(G)$  либо путем введения виртуального различия. Генерируемый граф задается числом вершин  $n$ , значением степени  $s$  для однородного графа, значениями  $(s_1, s_2, \dots)$  для неоднородного графа. При этом вершины с разными степенями генерируются примерно в равных долях.

Эксперименты проводились с графами, содержащими до 5000 вершин. Связность графов не контролировалась, т. е. могли встречаться и несвязные графы. Результаты экспериментов, отражающие время работы алгоритмов в миллисекун-

дах, сведены в таблицу. Для каждого значения  $n$  генерировалось два графа – однородный  $G_1$  со степенью  $s$  и неоднородный  $G_2$  со степенями  $s_1, s_2$ .

**Таблица.** Результаты экспериментов

**Table.** Experimental results

$n$	$G_1$		$G_2$		ISD-F		ISD-D		ISD-D		ISD-I		$\tau(A^2)$
	$s$	$s_1, s_2$	$P_w(G_1)$	$P_w(G_2)$	$P_G(G_1)$	$P_G(G_2)$	$P_G(H_1)$	$P_G(H_2)$	$P(G_1)$	$P(G_2)$	$P(G_1)$	$P(G_2)$	
50	5	5,10	1	0	1	0	1	1	1	6	4	0	
100	5	5,10	1	0	1	1	1	1	1	7	6	0	
200	10	10,20	2	1	2	1	2	1	8	6	0		
400	10	10,20	8	2	9	2	9	2	15	7	4		
600	15	15,30	25	3	26	3	27	3	44	10	25		
800	15	15,30	50	5	53	5	49	6	57	15	35		
1000	15	15,30	87	7	88	7	87	6	110	16	75		
1500	15	15,30	241	12	245	14	245	10	344	25	216		
2000	20	20,40	519	20	570	24	516	19	645	36	502		
2500	20	20,40	938	30	958	34	1042	27	1052	41	914		
3000	30	30,60	1708	46	1740	52	1704	40	1764	64	1612		
5000	30	30,60	7386	104	7526	124	7695	95	7674	131	7520		

Алгоритм ISD-F для графов  $G_1$  и  $G_2$  вычисляет полные инварианты  $P_w(G_1)$  и  $P_w(G_2)$ . Время работы алгоритма указано в соответствующих столбцах таблицы. При этом для однородного графа  $G_1$  начальное состояние  $D^0$  формировалось на основе маршрутной характеристики  $M^2(G_1)$ . При этом начальная дифференциация достигалась всегда и, следовательно, потребность введения виртуального различия не возникала. Время возведения матрицы в квадрат при вычислении маршрутной характеристики в таблице обозначено  $\tau(A^2)$ .

Работа алгоритма ISD-D представлена в таблице двумя экспериментами. Первый из них отражает работу алгоритма с графами  $G_1^*$  и  $G_2^*$ , полученными в результате перенумерации вершин в графах  $G_1$  и  $G_2$ . Изоморфизм графов в парах  $(G_1, G_1^*)$  и  $(G_2, G_2^*)$  алгоритм подтверждает, выполняя поиск полных инвариантов  $P_{G_1}(G_1^*)=P_w(G_1)$  и  $P_{G_2}(G_2^*)=P_w(G_2)$ . Затраты времени на проверку изоморфизма приведены в соответствующих столбцах таблицы. Для проведения второго эксперимента используются графы  $H_1$  и  $H_2$ , которые генерируются заново с тем же значением  $n$  и степенями, что и у графов  $G_1$  и  $G_2$ . В таблице приведены времена, затрачиваемые алгоритмом ISD-D на подтверждение того, что графы в парах  $(G_1, H_1)$  и  $(G_2, H_2)$  неизоморфны, т. е.  $P_{G_1}(H_1) \neq P_w(G_1)$  и  $P_{G_2}(H_2) \neq P_w(G_2)$ . Заметим, что в ходе экспериментов не было случаев, когда графы в этих парах оказывались изоморфными.

В таблице приведены также затраты времени на вычисление полных инвариантов  $P(G_1)$  и  $P(G_2)$  с помощью алгоритма ISD-I. В общем случае при наличии полного инварианта  $P(G)$  графа  $G$  проверка на изоморфизм графов  $G$  и  $H$  сводится к вычислению полного инварианта  $P(H)$  и сравнения его с  $P(G)$ . Поэтому затраты времени на вычисление  $P(H)$  можно рассматривать как затраты на определение изоморфизма двух графов с помощью алго-

ритма ISD-I. В нашем эксперименте время вычисления  $P(G_1)$  можно сравнивать с временем вычисления  $P_{G_1}(G_1^*)$  или с  $P_{G_1}(H_1)$ , а время вычисления  $P(G_2)$ , соответственно, с  $P_{G_2}(G_2^*)$  или с  $P_{G_2}(H_2)$ . Из таблицы следует, что такое сравнение складывается не в пользу алгоритма ISD-I. Это отмечалось и ранее, когда перечислялись преимущества алгоритмов ISD-F и ISD-D при проверке графов на изоморфизм.

Сравнивая алгоритмы ISD-I и ISD-F, ISD-D, следует помнить о принципиальном отличии полного инварианта  $P(G)$  от  $P_w(G)$ . Полный инвариант  $P_G(H)$  здесь не рассматривается, т. к. он дублирует  $P_w(G)$  и выделяется лишь при условии  $P_G(H)=P_w(G)$ , когда граф  $H$  изоморфен графу  $G$ . Полный инвариант  $P(G)$ , в отличие от  $P_w(G)$ , является глобальным идентификатором структуры графа  $G$ . В то время как  $P_w(G)$  может рассматриваться лишь в паре с системой кодирования  $W_G(S)$ , которая выступает в качестве эталона для графов изоморфных графу  $G$ . Пара  $P_w(G)$ ,  $W_G(S)$  отражает один из возможных маршрутов процесса дифференциации вершин графа  $G$ , выполняемой алгоритмом ISD-F. При этом разные пользователи для одного и того же графа  $G$  могут получить разные пары  $P_w(G)$ ,  $W_G(S)$ .

Отличия в применении полных инвариантов  $P(G)$  и  $P_w(G)$  для проверки изоморфизма поясним на примере решения следующей задачи. Задано множество графов  $\{G_q\}$ ,  $q=1,2,\dots,Q$  с равными степенными инвариантами. Требуется разбить это множество на подмножества изоморфных графов. Решение данной задачи с помощью алгоритма ISD-I сводится к вычислению полных инвариантов  $P(G_q)$  для всех графов  $G_q$  множества и распределения их по подмножествам с равными  $P(G_q)$ . Для алгоритмов ISD-F и ISD-D задача решается несколько сложнее. Сначала для графа  $G_1$  вычисляется  $P_w(G_1)$  и  $W_{G_1}(S)$ . Далее для графа  $G_2$  делается попытка найти  $P_{G_1}(G_2)=P_w(G_1)$ . Если это удастся, то графы  $G_1$  и  $G_2$  изоморфны и они включаются в одно подмножество. Если нет, то для графа  $G_2$  вычисляется  $P_w(G_2)$  и  $W_{G_2}(S)$ . Для графа  $G_3$  в зависимости от ситуации проверку на изоморфизм приходится делать по отношению к  $W_{G_1}(S)$  и  $W_{G_2}(S)$ . В общем случае для очередного графа  $G_q$  проверку на изоморфизм следует делать по отношению к сформированным ранее  $W_G(S)$ . В результате граф  $G_q$  войдет в одно из подмножеств либо станет эталоном для нового подмножества с системой кодирования  $W_{G_q}(S)$ .

На примере данной задачи очевидны преимущества алгоритма ISD-I при работе с библиотеками графов. В качестве библиотеки графов можно рас-

сматривать, например, множество известных молекулярных структур химических соединений [1–4]. Перед включением очередного графа  $G$  в библиотеку необходимо проверить, не является ли он одним из графов, включенных ранее. Для этого достаточно вычислить полный инвариант  $P(G)$  и сравнить его с инвариантами графов библиотеки. Что касается алгоритмов ISD-F и ISD-D, то они могут сыграть важную роль в исследованиях по решению проблем оценивания сходства структур графов.

#### Заключение

В традиционных подходах к анализу структурных свойств систем используются модели в виде графов. В данной статье мы имеем обратную задачу – необходимо анализировать структурные свойства структуры графа. Для решения этой задачи весьма многообещающим оказалось представление модели структуры графа в виде динамической системы. Модель включает сеть автоматов, отражающую структуру графа, и блок дифференциации вершин, реализующий правила переходов автоматов из одного состояния в другое. В процессе функционирования такой системы автоматы накапливают (интегрируют) информацию об отношениях между структурными различиями в графе. В итоге автоматы приобретают разные состояния, что соответствует дифференциации вершин.

На основе данной модели разработан алгоритм вычисления инвариантного описателя структуры (ISD) в модификации F – свободная интеграция, D – зависимая интеграция, I – независимая интеграция. Алгоритмы F и D решают проблему изоморфизма графов, оценивая совпадения траекторий смены состояний сети автоматов у сравниваемых графов. Алгоритм I вычисляет полный инвариант графа, и для установления изоморфизма графов достаточно сравнить их полные инварианты.

Таким образом, проблема изоморфизма легко решается как с помощью алгоритмов F и D, так и путем сравнения полных инвариантов, вычисленных по алгоритму I. Об этом свидетельствуют результаты экспериментальных исследований.

Что касается проблемы сходства структур, то заметные продвижения в её решении авторы связывают с анализом траекторий смены состояний сетей автоматов для сравниваемых структур и оцениванием их сходства.

*Работа выполнена в рамках государственного задания «Наука».*

#### СПИСОК ЛИТЕРАТУРЫ

1. Скворцова М.И. Математические модели и алгоритмы в исследованиях связи между структурой и свойствами органических соединений: автореф. дис. ... д-ра физ.-мат. наук. – М., 2007. – 38 с.
2. Кинг Р.Б. Химические приложения топологии и теории графов. – М.: Мир, 1987. – 560 с.
3. Baskin I., Skvortsova M. On the Basis of Invariants of Labeled Molecular Graphs // Chem. Inf. Comput. Sci. – 1995. – V. 35. – № 3. – P. 527–531.
4. Varmuza K. Chemometrics in Practical Applications. – Rijeka, Croatia: InTech, 2012. – 326 p.
5. Кузнецов В.Г. Графовая модель системы энергосберегающих мероприятий // ВЕЖПТ. – 2013. – № 8 (63). URL: <http://cyberleninka.ru>

- berleninka.ru/article/n/grafovaya-model-sistemy-energoberegayuschih-meropriyatiy (дата обращения: 01.11.2014).
6. Pande A., Gupta M., Tripathi A.K. Design pattern mining for GIS application using graph matching techniques // Computer Science and Information Technology (ICCSIT): 3<sup>rd</sup> IEEE International Conference. – 2010, 9–11 July. – V. 3. – P. 477–482. URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5564518&isnumber=5563521> (дата обращения: 01.11.2014).
  7. Зыков А.А. Основы теории графов. – М.: Вузовская книга, 2004. – 664 с.
  8. Пролубников А.В. О новом полном инварианте ациклических графов // ПДМ. Приложение. – 2010. – № 3. URL: <http://cyberleninka.ru/article/n/o-novom-polnom-invariante-atsiklicheskih-grafov> (дата обращения: 04.11.2014).
  9. Lindell S.A. Logspace Algorithm for Tree Canonization // Proc. of the 24<sup>th</sup> Annual ACM Symposium on the Theory of Computing. – New York, 1992. – P. 400–404.
  10. Planar Graph Isomorphism is in Log-Space. 24<sup>th</sup> Annual IEEE Conference on Computational Complexity / S. Datta, N. Limaye, P. Nimbhorkar, T. Thierauf, F. Wagner. – Paris, 15–18 July, 2009. – P. 203–214.
  11. Пономаренко И.Н. Проблема изоморфизма графов: Алгоритмические аспекты (Записки к лекциям). Санкт-Петербургское отделение Математического института им. В.А. Стеклова. Октябрь-декабрь 2010. URL: [http://logic.pdmi.ras.ru/csclub/sites/default/files/graph\\_isomorphism\\_ponomarenko\\_lecture\\_notes.pdf](http://logic.pdmi.ras.ru/csclub/sites/default/files/graph_isomorphism_ponomarenko_lecture_notes.pdf) (дата обращения 04.11.2014).
  12. Trofimov M.I. Polynomial Time Algorithm for Graph Isomorphism Testing. – 18 Jun 2013. URL: <http://arxiv.org/pdf/1004.1808v6.pdf> (дата обращения 04.11.2014).
  13. Вильховский Д.Э., Ефимов С.С. Исследование алгоритмов определения изоморфности графов в системах с распределённой и общей памятью // Математические структуры и моделирование. – 2012. – Вып. 26. – С. 89–93.
  14. Новый функционал информативности для анализа структуры химических графов / М. Дэмер, Ф. Эммерт-Штрайб, Ю.П. Цой, К. Вармуза // Известия Томского политехнического университета. – 2010. – Т. 316. – № 5. – С. 5–11.
  15. Quantum Frontiers of Atoms and Molecules / M. Dehmer, F. Emmert-Streib, R.Y. Tsoy, K. Varmuza. – New York: Nova Publishing, 2011. – 673 p.
  16. Emmert-Streib F., Dehmer M. Networks for systems biology: conceptual connection of data and function // IET Systems Biology. – 2011. – V. 5. – № 3. – P. 185–207.
  17. Dehmer M., Grabner M. The Discrimination Power of Molecular Identification Numbers Revisited // MATCH Commun. Math. Comput. Chem. – 2013. – V. 69. – № 3. – P. 785–794.
  18. De Matos S., Dehmer M., Emmert-Streib F. Interfacing cellular networks of *S. cerevisiae* and *E. coli*: Connecting dynamic and genetic information // BMC Genomics. – 2013. – V. 14. – № 324. – P. 111–134.
  19. An efficient heuristic approach to detecting graph isomorphism based on combinations of highly discriminating invariants / M. Dehmer, M. Grabner, A. Mowshowitz, F. Emmert-Streib // Advances in Computational Mathematics. – 2013. – V. 39. – № 2. – P. 311–325.
  20. McKay B.D., Piperno A. Practical Graph Isomorphism // J. Symbolic Computation. – January 2014. – V. 60. – P. 94–112.
  21. Balasubramanian K., Parthasarathy K.R. In search of a complete invariant for graphs // Lect. Notes Math. – 1981. – V. 885. – P. 42–59.
  22. Погребной В.К., Погребной Ан.В. Полиномиальный алгоритм вычисления полного инварианта графа на основе интегрального описателя структуры // Известия Томского политехнического университета. – 2013. – Т. 323. – № 5. – С. 152–159.
  23. 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.
  24. Погребной Ан.В. Полный инвариант графа и алгоритм его вычисления // Известия Томского политехнического университета. – 2014. – Т. 325. – № 5. – С. 110–122.

Поступила 06.11.2014 г.

UDC 519.171.1

## METHOD OF GRAPH VERTICES DIFFERENTIATION AND SOLUTION OF THE ISOMORPHISM PROBLEM

**Andrey V. Pogrebnoy,**

National Research Tomsk Polytechnic University, 30, Lenin Avenue, Tomsk, 634050, Russia. E-mail: avpogrebnoy@gmail.com

**Vladimir K. Pogrebnoy,**

National Research Tomsk Polytechnic University, 30, Lenin Avenue, Tomsk, 634050, Russia. E-mail: vkp@tpu.ru

*Modelling the relations between the structure of the object and its properties is one of the current problems in the modern applied graph theory. The problem of identification and evaluation of structures similarity plays the important role in solving this task. The known approaches to the structures similarity estimation in power engineering, geographic information systems, computer chemistry, including petrochemistry, are limited by using a set of indirect properties and do not consider the possibility of solving the structures similarity problem with the help of direct properties, associated with isomorphism.*

**The main aim of the study** is to state the theoretical bases of graph vertices differentiation method and to show the method possible applications for estimating the structures similarity and for solving the isomorphism problem, considering it as a special case of complete structures similarity.

**The methods used in the study** are based on the applied graph theory, efficient algorithms analysis and construction theory, simulating structures using automata models and application of theory of integration of codes of structural differences.



**The results.** The authors have stated the problem of graph structure identification, which unites invariant description, complete invariant, graph isomorphism and structures similarity problems. It is shown that the solution of these problems is reduced in general to solving vertices differentiation problem. The paper introduces three types of structural differences in a graph – basic, hidden and virtual. The authors offered a graph structure automata model as the basis of the vertices differentiation method and development of complete graph invariant computation algorithms with free (algorithm ISD-F), dependent (ISD-D) and independent (ISD-I) integration of structure differences codes. The paper considers the features of these algorithms application for solving the isomorphism problem and the possibility of developing the algorithm for evaluating the structures similarity based on the interdependent codes integration. The authors carried out the experimental studies of algorithms for computing complete invariant for graphs containing up to 5000 vertices. The experiments have shown high efficiency of these algorithms.

**Key words:**

Graph structure identification, complete graph invariant, graph isomorphism, integral structure descriptor, structure automata model, graph vertices differentiation.

The research was carried out within the state task «Nauka».

**REFERENCES**

1. Skvorcova M.I. *Matematicheskie modeli i algoritmy v issledovaniiakh svyazi mezhdu strukturoy i svoystvami organicheskikh soedineniy. Avtoreferat Dis. Dokt. nauk* [Mathematical models and algorithms in research of connection between structure and properties of organic compounds. Dr. Diss. Abstract]. Moscow, 2007. 38 p.
2. King R.B. *Khimicheskie prilozheniya topologii i teorii grafov* [Chemical applications of graphs topology and theory]. Moscow, Mir Publ., 1987. 560 p.
3. Baskin I., Skvortsova M. On the Basis of Invariants of Labeled Molecular Graphs. *Chem. Inf. Comput. Sci.*, 1995, vol. 35, no. 3, pp. 527–531.
4. Varmuza K. *Chemometrics in Practical Applications*. Rijeka, Croatia, InTech, 2012. 326 p.
5. Kuznetsov V.G. Grafovaya model sistemy energosberegashchikh meropriyatii [Graph model of energy-saving measures]. *Eastern-European Journal of Enterprise Technologies*, 2013, vol. 63, no. 8. Available at: <http://cyberleninka.ru/article/n/grafovaya-model-sistemy-energoberegayuschih-meropriyatii> (accessed 01 November 2014).
6. Pande A., Gupta M., Tripathi A.K. Design pattern mining for GIS application using graph matching techniques. *Computer Science and Information Technology (ICCSIT)*. 3<sup>rd</sup> IEEE International Conference, 2010, 9–11 July, vol. 3, pp. 477–482. Available at: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5564518&isnumber=5563521> (accessed 01 November 2014).
7. Zykov A.A. *Osnovy teorii grafov* [Basics of the graph theory]. Moscow, Vuzovskaya kniga Publ., 2004. 664 p.
8. Prolubnikov A.V. O novom polnom ivariante atsiklicheskikh grafov [On a new complete acyclic graph invariant]. *PDM. Prilozhenie*, 2010, no. 3. Available at: <http://cyberleninka.ru/article/n/o-novom-polnom-invariante-atsiklicheskikh-grafov> (accessed 14 November 2014).
9. Lindell S.A. Logspace Algorithm for Tree Canonization. *Proc. of the 24<sup>th</sup> Annual ACM Symposium on the Theory of Computing*. New York, 1992. pp. 400–404.
10. Datta S., Limaye N., Nimbhorkar P., Thierauf T., Wagner F. Planar Graph Isomorphism is in Log-Space. *24<sup>th</sup> Annual IEEE Conference on Computational Complexity*. Paris, 15–18 July, 2009. pp. 203–214.
11. Ponomarenko I.N. *Problema izomorfizma grafov: Algoritmicheskie aspekty (Zapiski k lekciyam)* [Problem of graph isomorphism: Algorithmic aspects (Lecture notes)]. St. Petersburg Department of Steklov Mathematical Institute RAS. October–December 2010. Available at: [http://logic.pdmi.ras.ru/csclub/sites/default/files/graph\\_isomorphism\\_ponomarenko\\_lecture\\_notes.pdf](http://logic.pdmi.ras.ru/csclub/sites/default/files/graph_isomorphism_ponomarenko_lecture_notes.pdf) (accessed 04 November 2014).
12. Trofimov M.I. *Polynomial Time Algorithm for Graph Isomorphism Testing*. 18 June 2013. Available at: <http://arxiv.org/pdf/1004.1808v6.pdf> (accessed 04 November 2014).
13. Vilkhovskiy D.E., Efimov S.S. Issledovanie algoritmov opredeleniya izomorfnosti grafov v sistemakh s raspredelennoy i obshchey pamatyu [Research of graph isomorphism determination algorithms in the systems with distributed memory]. *Mathematical Structures and Modeling*, 2012, vol. 26, pp. 89–93.
14. Demer M., Emmert-Shtrayb F., Tsou Yu.P., Varmuza K. Noviy funkcional informativnosti dlya analiza struktury khimicheskikh grafov [New informative functional for chemical graph structure analysis]. *Bulletin of the Tomsk Polytechnic University*, 2010, vol. 316, no. 5, pp. 5–11.
15. Dehmer M., Emmert-Streib F., Tsou R.Y., Varmuza K. *Quantum Frontiers of Atoms and Molecules*. New York, Nova Publishing, 2011. 673 p.
16. Emmert-Streib F., Dehmer M. Networks for systems biology: conceptual connection of data and function. *IET Systems Biology*, 2011, vol. 5, no. 3, pp. 185–207.
17. Dehmer M., Grabner M. The Discrimination Power of Molecular Identification Numbers Revisited. *MATCH Commun. Math. Comput. Chem.*, 2013, vol. 69, no. 3, pp. 785–794.
18. De Matos S., Dehmer M., Emmert-Streib F. Interfacing cellular networks of *S. cerevisiae* and *E. coli*: Connecting dynamic and genetic information. *BMC Genomics*. 2013, vol. 14, no. 324, pp. 111–134.
19. Dehmer M., Grabner M., Mowshowitz A., Emmert-Streib F. An efficient heuristic approach to detecting graph isomorphism based on combinations of highly discriminating invariants. *Advances in Computational Mathematics*, 2013, vol. 39, no. 2, pp. 311–325.
20. McKay B.D., Piperno A. Practical Graph Isomorphism. *J. Symbolic Computation*, January 2014, vol. 60, pp. 94–112.
21. Balasubramanian K., Parthasarathy K.R. In search of a complete invariant for graphs. *Lect. Notes Math.*, 1981, vol. 885, pp. 42–59.
22. Pogrebnoy V.K., Pogrebnoy An.V. Polinomialny algoritm vychisleniya polnogo invarianta grafa na osnove integralnogo opisatelya struktury [Polynomial algorithm of complete graph invariant computation on the basis of integral structure descriptor]. *Bulletin of the Tomsk Polytechnic University*, 2013, vol. 323, no. 5, pp. 152–159.
23. Pogrebnoy V.K., Pogrebnoy An.V. Issledovanie polinomialnosti metoda vychisleniya integralnogo opisatelya struktury grafa [Research of the integral structure descriptor computation method polynomiality]. *Bulletin of the Tomsk Polytechnic University*, 2013, vol. 323, no. 5, pp. 146–151.
24. Pogrebnoy An.V. Polniy invariant grafa i algoritm ego vychisleniya [Complete graph invariant and algorithm of its computation]. *Bulletin of the Tomsk Polytechnic University*, 2014, vol. 325, no. 5, pp. 110–122.

Received: 06 November 2014.