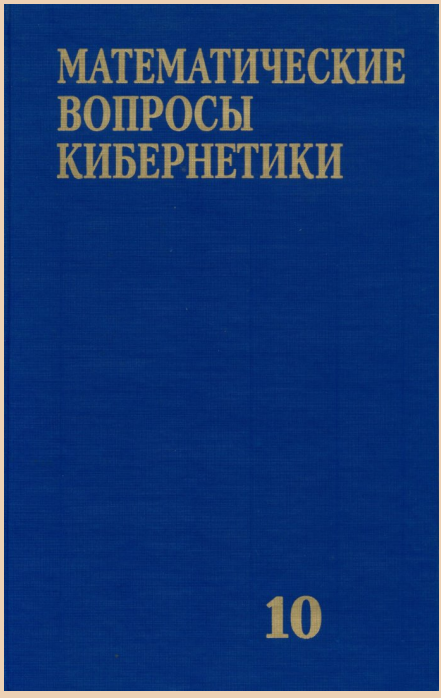
ИПМ им. М.В. Келдыша РАН • Электронная библиотека
Математические вопросы кибернетики • Выпуск 10



Э. Э. Гасанов

Информационно-графовая модель в теории баз данных

Рекомендуемая форма библиографической ссылки: Гасанов Э. Э. Информационно-графовая модель в теории баз данных // Математические вопросы кибернетики. Вып. 10. — М.: ФИЗМАТЛИТ, 2001. — С. 225–234. URL: http://library.keldysh.ru/mvk.asp?id=2001-225

ИНФОРМАЦИОННО-ГРАФОВАЯ МОДЕЛЬ В ТЕОРИИ БАЗ ДАННЫХ*)

Э. Э. ГАСАНОВ

(МОСКВА)

База данных (БД) — формализованное представление информации, удобное для хранения и поиска данных в нем. Понятие БД возникло в 60-е годы 20 века и связано с развитием вычислительной техники и информатики. Тематика теории БД связана с поиском удобного представления, компактного хранения, быстрого поиска, защищенности и других свойств данных. Развитию этого направления способствовали как производственные и творческие коллективы: IBM (иерархическая модель данных), Рабочая группа по БД Ассоциации по языкам систем обработки данных — CODASYL (сетевая модель данных), исследовательская группа по системам управления БД Американского национального института стандартов — ANSI/SPARC Study Group on DataBase Management Systems (принципы проектирования БД), так и отдельные исследователи: Кодд (E. F. Codd) (реляционная модель данных), Галлейр (H. Gallaire), Минкер (J. Minker) (дедуктивные БД), Тальхайм (B. Thalheim) (моделирование семантики в БД, теория ограничений целостности), Аткинсон (M. Atkinson), Бири (C. Beeri) и др. (объектно-ориентированные БД), В. Н. Решетников (алгебраическая модель информационного поиска), Думи (A. Dumey), А. П. Ершов (методы хеширования), Бентли (J. L. Bentley), Кнут (D. E. Knuth), Ли (D. T. Lee), Маурер (W. D. Maurer), Ульман (J. D. Ullman), Шеймос (M. I. Shamos) и др. (сложность алгоритмов обработки данных), Э. Э. Гасанов (информационно-графовая модель данных, сложность алгоритмов поиска) и многие другие.

Основные виды представления (модели) данных сформировались под влиянием практики с использованием средств математики.

В основу *реляционной модели данных* [9] положено понятие отношения. Подмножество $R \subseteq D_1 \times D_2 \times \cdots \times D_n$ есть отношение арности n с доменами $D_1, D_2, \ldots, D_n$. Рассматриваемые отношения, как правило, конечные, поэтому удобно представлять их в виде плоских таблиц. Строки таблицы называются кортежами, а столбцы — атрибутами. Подмножество атрибутов отношения называется ключом, если проекция таблицы на это множество состоит из разных строк, но удаление любого атрибута из ключа нарушает это свойство. Понятие ключа соответствует тупиковому тесту, поэтому результаты, полученные в теории тестов (А. Е. Андреев, В. Н. Носков, Г. Р. Погосян и др.) могут быть использованы для оценки мощности ключа и минимального числа атрибутов в ключе. Оценивались мощности ключей

*) Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект 01-01-00748).

в случайных БД (таблицах) (О. Б. Селезнев, Б. Тальхайм и др.). Реляционная БД — это множество таблиц, обогащенное операциями объединения, пересечения, разности, декартова произведения, проекции, соединения, селекции, частного и др. Изучение этих алгебр отношений составляет содержание теории реляционных БД. Ассоциируемый с *табличным представлением* способ хранения данных не компактен, а способ поиска — полный перебор. В связи с этим в современных реляционных БД реляционная модель используется только как внешнее представление, т. е. представление пользователя, тогда как внутреннее представление данных, т. е. представление на машинных носителях, — принципиально другое. Простота представления при сохранении всех функциональных возможностей БД делают реляционную модель удобной для изучения качественных свойств и характеристик БД. Наглядность и простота понимания — причина популярности этой модели среди большинства пользователей.

Если в табличном представлении произвести лексикографическое упорядочение и сделать склейку по совпадающим префиксам, то получается *древовидное представление данных*, в основе которого лежит понятие дерева. Такое представление ведет к большей компактности данных и к ускорению поиска нужных данных. Такие древовидные структуры, как бинарные деревья, 2-3-деревья, B-деревья, сортирующие деревья [1] используются для внутреннего представления данных. Древовидное представление удобно использовать в лингвистических и подобных им БД, например, когда надо найти то или иное слово. В случае поиска множества однокоренных слов, то есть слов с заданной средней частью, древовидное представление не очень удобно. Древовидное представление все еще достаточно просто для понимания, хотя и не так наглядно как табличное. При использовании *древовидной (или иерархической) модели данных* как внешнего представления данных предполагаются иерархические отношения между данными, т. е. отношения типа родитель-потомки, когда у каждого объекта только один родитель, но может быть несколько потомков. Одной из систем, использующих иерархическую модель данных, является система IMS фирмы IBM [22].

Обобщение понятия дерева до графа аналогично переходу от древовидного к *сетевому представлению данных*. При векторном виде данных в древовидном представлении склейка данных может быть сделана только по начальному отрезку; в сетевом представлении она допустима по любым отрезкам. В *сетевой модели данных* доступ к данным может быть осуществлен по многим путям. Она позволяет реализовать более широкий класс отношений между объектами, чем древовидная. Эта модель данных развивается ассоциацией CODASYL [12]. Поскольку сетевая модель является обобщением древовидной, то она предоставляет больше возможностей как для описания предметной области, представляемой БД, так и для нахождения оптимальных решений хранения и поиска данных. Но использование сетевой модели требует высокой квалификации от разработчика и поэтому она не была воспринята массовым пользователем.

В *объектно-ориентированной модели данных* [13], опирающейся на принципы объектно-ориентированного программирования, каждый объект представляется как черный ящик, доступ к данным которого осуществляется только через специальные функции, прилагаемые к нему. Из таких ящиков строятся более сложные объекты, которые в свою очередь могут служить новыми ящиками для построения объектов следующего уровня сложности и т. д. Главное преимущество объектно-ориентированной модели — ее технологичность, в связи с чем это одна из наиболее развивающихся моделей сегодня.

Под влиянием математической логики строится *дедуктивная модель данных* [21]. Данные в дедуктивных базах данных рассматриваются как аксиомы, а новые данные получаются из аксиом путем логического вывода. Преимущество этой модели — компактность начального множества данных (все зашито в аксиомы и правила вывода) и потенциальная бесконечность множества выводимых фактов. Недостаток — большие ресурсы по времени и памяти, необходимые для процесса вывода. Дедуктивные базы данных удобно использовать в системах принятия решений, когда заранее не очерчена область возможных ситуаций.

Модели данных для описания внутренней (или физической) организации БД имеют дело с размещением данных на запоминающих устройствах. Здесь главным является эффективность функционирования БД, то есть обеспечение эффективности поиска, занесения, удаления и компактность хранения данных. При этом требования быстроты вставки, поиска и компактности хранения данных находятся в противоречии. Для быстрого поиска необходимо строить дополнительные сложные структуры данных, которые затрудняют процесс вставки новых данных и не способствуют компактности. Одной из моделей, используемых для исследования сложности алгоритмов, является алгебраическое дерево вычислений (АДВ-модель) [1], с помощью которой получены оценки сложности алгоритмов работы с данными.

Моделью данных, предназначенной для исследования сложностных характеристик алгоритмов работы с данными, является информационно-графовая модель [4, 5], основывающаяся на теории управляющих систем.

Информационно-графовая модель данных (ИГМД) — модель данных, основанная на представлении данных в виде функционально нагруженного графа, который, определяя по запросу вычислительный процесс, выдает нужный ответ.

Если X — множество символов запросов с заданным на нем вероятностным пространством $\langle X, \sigma, P\rangle$, где σ — алгебра подмножеств множества X, P — вероятностная мера на σ; Y — множество символов данных (записей); ρ — бинарное отношение на $X \times Y$, называемое отношением поиска; то пятерка $S = \langle X, Y, \rho, \sigma, P\rangle$ называется *типом*. Тройка $I = \langle X, V, \rho\rangle$, где V — некоторое конечное подмножество множества Y, называемое библиотекой, называется задачей информационного поиска (ЗИП) типа S. Содержательно ЗИП $I = \langle X, V, \rho\rangle$ состоит в перечислении для произвольно взятого запроса $x \in X$ всех и точно тех записей $y \in V$, что $x\rho y$. Если F и G — суть множества символов одноместных предикатов и переключателей соответственно, определенных на X, где переключатель — функция, областью значений которой является конечное подмножество натурального ряда, то пара $\mathcal{F} = \langle F, G\rangle$ называется базовым множеством и описывает множество элементарных операций, используемых при решении задачи информационного поиска.

Над базовым множеством $\mathcal{F} = \langle F, G\rangle$ определяется понятие *информационного графа* (ИГ). В конечной многополюсной ориентированной сети выбирается вершина — полюс, называемая корнем. Остальные полюса называются листьями и им приписываются записи из Y. Некоторые вершины сети называются переключательными и им приписываются переключатели из G. Ребра, исходящие из каждой из переключательных вершин, нумеруются и называются переключательными ребрами. Ребра, не являющиеся переключательными, называются предикатными, и им приписываются предикаты из множества F. Таким образом нагруженную многополюсную ориентированную сеть называют информационным графом над базовым множеством $\mathcal{F} = \langle F, G\rangle$. Затем определяется *функционирование ИГ*. Предикатное ребро проводит запрос $x \in X$, если предикат ребра истинен на x; переклю-

чательное ребро под номером n проводит x, если переключатель начала этого ребра равен n на x; ориентированная цепочка ребер проводит x, если каждое ребро цепочки проводит x; запрос x проходит в вершину β ИГ, если существует ориентированная цепь, ведущая из корня в вершину β, которая проводит x; запись y, приписанная листу α, попадает в ответ ИГ на x, если x проходит в лист α. Ответом ИГ U на запрос x называют множество записей, попавших в ответ U на x, и обозначают его $\mathcal{J}_U(x)$. Эту функцию $\mathcal{J}_U(x)$ считают результатом функционирования ИГ U. ИГ наряду со структурой данных описывает алгоритм соответствующего поиска. Процесс поиска при заданном запросе начинается с корня и распространяется в зависимости от нагрузочных функций, возможно, сразу по нескольким направлениям. Если этот процесс на графе достигает элементов данных, то они включаются в ответ алгоритма.

ИГ U *разрешает* ЗИП $I = \langle X, V, \rho \rangle$, если $\mathcal{J}_U(x) = \{y \in V\colon x\rho y\}$. Вводится *сложность ИГ*. Предикат $\varphi_\beta(x)$ истинный на x, если x проходит в вершину β, и ложный в противном случае, называется функцией фильтра вершины β. *Сложностью ИГ* U *на запросе* $x \in X$ называется число $T(U, x) = \sum\limits_{\beta \in \mathcal{P}} \varphi_\beta(x) + \sum\limits_{\beta \in \mathcal{R} \setminus \mathcal{P}} \psi_\beta \cdot \varphi_\beta(x)$, где $\mathcal{R}$ — множество вершин ИГ U, $\mathcal{P}$ — множество переключательных вершин U, ψ_β — количество ребер, исходящих из вершины β. Эта величина равна числу функций, вычисленных алгоритмом поиска, определяемым ИГ U, на запросе x.

Если каждая функция из $\mathcal{F}$ — измерима (относительно алгебры σ), то для любого ИГ U над $\mathcal{F}$ функция $T(U, x)$ измерима. *Сложностью ИГ* U называется математическое ожидание величины $T(U, x)$, равное $T(U) = M_x T(U, x)$. Она характеризует среднее время поиска. *В-сложностью ИГ* U называется число $\overline{T}(U) = \max\limits_{x \in X} T(U, x)$. Эта величина характеризует время поиска в худшем случае. *Объемом ИГ* U называется число $Q(U)$ ребер U. Он характеризует объем памяти, необходимый для хранения данных. Пусть нам дана некая ЗИП I. *Сложностью задачи* I *при базовом множестве* $\mathcal{F}$ *и заданном объеме* q назовем число

$$T(I, \mathcal{F}, q) = \inf\{T(U)\colon U \in \mathcal{U}(I, \mathcal{F}) \text{ и } Q(U) \leqslant q\},$$

где $\mathcal{U}(I, \mathcal{F})$ — множество всех ИГ над базовым множеством $\mathcal{F}$, разрешающих ЗИП I. Число

$$T(I, \mathcal{F}) = \inf\{T(U)\colon U \in \mathcal{U}(I, \mathcal{F})\}$$

назовем *сложностью задачи* I *при базовом множестве* $\mathcal{F}$.

Особенностью ИГМД является удобство оценивания среднего времени работы алгоритма.

Работа с БД предполагает создание удобных языков — *языков манипулирования данными*, — примеры которых доставляют формальные языки логики и алгебры. В алгебраических языках манипулирования данными запрос к БД определяет последовательность операций, которые приведут к ответу. Такими языками являются ISDL и АСТРИД [9, 11]. В языках манипулирования данными, основанных на исчислении предикатов, запрос к БД соответствует формуле некоторой формально-логической теории, а ответом является множество объектов из области интерпретации, на котором истинна формула, соответствующая запросу. Такими языками являются QUEL, SQL, QBE [9, 11].

Тематика теории БД связана с двумя основными направлениями. Первое в основном опирается на аппарат формально-логических теорий и связано с изучением качественных свойств моделей данных и их семантическим

обоснованием. К этому же направлению относятся вопросы выразительности и сложности языков запросов, соответствующих моделям данных, изучение БД как теорий (например, в интуиционистской логике высшего порядка), а также *теория ограничений целостности* [25]. В ней изучаются ограничения, накладываемые предметной областью, определяющие связи между компонентами БД и описывающие поведение БД во времени. Ограничения целостности могут быть использованы в качестве языка описания семантики БД. Изучаются особого вида ограничения целостности, называемые зависимостями. Они делятся на статические (отражающие семантику всех возможных состояний БД) и динамические (описывающие поведение БД во времени, т. е. корректность последовательностей ее состояний). Изучается проблема выводимости некоторой данной зависимости из заданного списка зависимостей. Выделены классы зависимостей, в которых эта проблема неразрешима и P- или NP-разрешима. При разрешимости проблемы выводимости актуальной становится задача об аксиоматизации соответствующего класса зависимостей. Известны случаи аксиоматизируемых и неаксиоматизируемых классов.

Второе основное направление тематики теории БД связано с вопросами сложности алгоритмов обработки данных. Практика выявила несколько типов задач информационного поиска, для эффективной реализации которых было разработано большое количество алгоритмов с разными соотношениями таких характеристик алгоритмов как время работы в худшем случае и в среднем, объем памяти, необходимый алгоритму. Каждый из этих алгоритмов имеет свою структуру хранения данных, и соответствующие ей алгоритмы обновления данных. Среди основных типов задач информационного поиска можно выделить следующие. *Задача поиска идентичных объектов*, когда в множестве данных надо найти объекты равные запросу. *Задача о близости*, которая состоит в поиске во множестве, в котором задан линейный порядок, объекта, ближайшего к объекту-запросу. *Задача включающего поиска* (или дескрипторный поиск, или поиск по ключевым словам), когда запрос задает набор свойств и надо найти объекты, обладающие этими свойствами. *Задача о доминировании*, которая состоит в поиске в конечном подмножестве n-мерного пространства всех тех точек, которые не больше по каждой из компонент, чем запрос, являющийся в данном случае точкой n-мерного пространства. *Задача интервального поиска*, которая состоит в поиске в конечном подмножестве n-мерного пространства всех тех точек, которые попадают в n-мерный параллелепипед-запрос.

Для любой задачи информационного поиска справедлива элементарная нижняя оценка сложности, говорящая, что время поиска не меньше чем время перечисления ответа (эта оценка называется мощностной).

Для задачи поиска идентичных объектов с помощью деления пополам легко получается логарифмическая верхняя оценка времени поиска в худшем случае, с другой стороны для этой задачи в рамках АДВ-модели получена логарифмическая нижняя оценка времени поиска в худшем случае (теоретико-информационная нижняя оценка сложности, А. Б. Бородин [19]). С помощью методов хеширования получены алгоритмы поиска идентичных объектов, для которых доказано (А. П. Ершов [8]), что в среднем время поиска равно константе, но в худшем случае эти алгоритмы равны перебору, то есть имеют линейную сложность. Аналогичные результаты получаются для задачи о близости.

Задачу о доминировании и задачу интервального поиска относят также к направлению, называемому *вычислительная геометрия* [10] и посвященному исследованию сложности алгоритмов решения геометрических задач. По задаче интервального поиска получены следующие результаты. Бентли (J. L. Bentley) в [14] предложил метод многомерного двоичного дерева (k-D-дерева), исходя из того, что бинарное дерево для одномерной задачи

дает хороший результат. Этот метод имеет линейные затраты по памяти, но Ли (D. T. Lee) и Вонг (C. K. Wong) [23] показали, что в худшем случае этим алгоритмом двумерная задача решается за время $O(\sqrt{k})$. Здесь и далее, k равно мощности библиотеки, а оценки приводятся без времени перечисления ответа. В работе [15] Бентли предложил метод прямого доступа, который решает задачу за время $O(\log k)$, но требует затрат по памяти порядка k^3. Чтобы уменьшить требуемую память в работе [16] Бентли и Маурером (W. D. Maurer) был предложен многоэтапный метод прямого доступа. Он позволяет снижать порядок требуемой памяти, но при этом возрастает константа при логарифме в оценке времени. С помощью метода дерева интервалов (или дерева регионов) Бентли и Шеймос (M. I. Shamos) [17] получили оценку времени поиска $O(\log^2 k)$ при затратах памяти $O(k \log k)$. Уиллард (D. E. Willard) [26] и Люкер (G. S. Lueker) [24] независимо предложили модификацию дерева интервалов, которая позволила снизить время поиска до $O(\log k)$ при тех же затратах памяти. Чазелле (B. M. Chazelle) в [20] смог снизить затраты памяти до $O(k \log k/\log\log k)$, но при этом возросла константа при логарифме в оценке времени. Во всех этих работах оценивается время поиска в худшем случае. Болоур в [18] описал метод хеширования, который он использовал вместо поиска по древовидным структурам в задаче интервального поиска. Применение этого метода позволило получить довольно быстрые в среднем решения задачи интервального поиска при условии, что область запросов находится в заранее определенных границах.

ИГМД позволяет взглянуть на эту разрозненную картину с единых позиций. Пусть дана некоторая ЗИП $I = \langle X, V, \rho\rangle$. Она задает множество предикатов на X

$$C_{V,\rho} = \{\chi_{y,\rho}(x) = \begin{cases} 1, & \text{если } x\rho y \\ 0 & \text{в противном случае} \end{cases} : y \in V\}.$$

Задача построения хорошего алгоритма поиска, решающего данную ЗИП, сводится к построению схемы, реализующей как функции проводимости функции из множества $C_{V,\rho}$, где в качестве схем удобно рассматривать ИГ. После введения сложности схемы мы получаем стандартную для теории синтеза управляющих систем постановку: по заданной системе функций (задаваемой ЗИП) построить схему (информационный граф), реализующую эту систему функций и имеющую минимальную сложность. Таким образом, ИГМД позволяет решать вопросы построения хороших алгоритмов поиска методами теории управляющих систем. В рамках информационно-графовой модели получены следующие результаты.

Задача поиска идентичных объектов состоит в поиске в множестве объекта, идентичного объекту-запросу, и формально принадлежит типу $S_{id} = \langle (0, 1], (0, 1], =, \sigma, P\rangle$. *Задача о близости* состоит в поиске в линейно-упорядоченном множестве объекта, ближайшего к объекту-запросу, и принадлежит типу $S_{ne} = \langle (0, 1], (0, 1], \rho_{ne}, \sigma, P\rangle$, где ρ_{ne} задается на $(0, 1] \times V$ и определяется соотношением $x\rho_{ne}y \Longleftrightarrow (y \in V)\,\&\,(x \leqslant y)\,\&\,(\neg(\exists y')((y' \in V)\,\&\,\&\,(x \leqslant y')\,\&\,(y' < y)))$.

Пусть

$$F_1 = \{f_{=,a}(x) = \begin{cases} 0, & \text{если } x \neq a \\ 1, & \text{если } x = a \end{cases} : a \in (0, 1]\}, \tag{1}$$

$$G_1 = \{g_{\leqslant,a}(x) = \begin{cases} 1, & \text{если } x \leqslant a \\ 2 & \text{в противном случае} \end{cases} : a \in (0, 1]\}, \tag{2}$$

$$G_2 = \{]x \cdot m[: m = 1, 2, 3\ldots\},\ \mathscr{F}_1 = \langle F_1, G_1 \cup G_2\rangle. \tag{3}$$

Справедлива теорема [2].

Теорема 1. *Пусть вероятностная мера P определяется ограниченной функцией плотности распределения, I — ЗИП типа S_{id} или типа S_{ne}, $\mathcal{F}_1$ — базовое множество, задаваемое соотношениями* (1)–(3). *Тогда* $1 < T(I, \mathcal{F}_1) < 2$.

Также получены алгоритмы, константные по времени в худшем случае при квадратичных затратах памяти [7].

Пусть X — множество запросов, Y — множество записей, ρ — отношение поиска на $X \times Y$, $y \in Y$, $V \subset Y$, $I = \langle X, V, \rho \rangle$. Тогда обозначим $O(y, \rho) = \{x \in X: x \rho y\}$, $R(I) = \sum\limits_{y \in V} P(O(y, \rho))$. Последняя величина равна средней длине ответа, или среднему времени перечисления ответа.

Задача включающего поиска принадлежит $S_b = \langle B^n, B^n, \succeq, \sigma, P \rangle$, где B^n — *единичный n-мерный куб*, $\succeq$ — отношение частичного порядка на B^n «не меньше по-компонентно», P — равномерная вероятностная мера на B^n. Справедлива теорема [3].

Теорема 2. *Пусть базовое множество имеет вид $\mathcal{F} = \langle F, \varnothing \rangle$, где $F \subseteq \mathcal{M}^n$ и $\mathcal{K}^n \subseteq F$, и $\mathcal{M}^n$ — множество монотонных булевых функций, а $\mathcal{K}^n$ — множество элементарных монотонных конъюнкций. Тогда для любой ЗИП I типа S_b справедливо неравенство $T(I, \mathcal{F}) \geqslant 2R(I)$ и существуют такие ЗИП I типа S_b, что $T(I, \mathcal{F}) = 2R(I)(1 + \bar{o}(1))$ при $n \to \infty$.*

Задача о доминировании состоит в поиске в конечном подмножестве n-мерного пространства всех тех точек, которые не больше по каждой из компонент чем запрос, являющийся в данном случае точкой n-мерного пространства. Пусть $X_n = (0, 1]^n$. Отношение поиска ρ_1 определено на $X_n \times X_n$ и задается следующим соотношением $(x_1, x_2, \ldots, x_n)\rho_1(y_1, y_2, \ldots, y_n) \Longleftrightarrow y_i \leqslant x_i$, $i = 1, 2, \ldots, n$. Тогда тип $S_d = \langle X_n, X_n, \rho_1, \sigma, P \rangle$ назовем типом задачи о доминировании. Пусть

$$G_3 = \{g_{i, \cdot, m}(x_1, \ldots, x_n) =]x_i \cdot m[: i \in \{1, 2, \ldots, n-1\}, \ m = 1, 2, 3 \ldots\}, \quad (4)$$

$$G_4 = \{g_{i, <, a}(x_1, \ldots, x_n) = \begin{cases} 1, & \text{если } x_i < a \\ 2, & \text{если } x_i \geqslant a \end{cases} : i \in \{1, 2, \ldots, n-1\}, \ a \in (0, 1]\}, \quad (5)$$

$$F_2 = \{g_{n, \geqslant, a}(x_1, \ldots, x_n) = \begin{cases} 0, & \text{если } x_n < a \\ 1, & \text{если } x_n \geqslant a \end{cases} : a \in (0, 1]\}, \quad (6)$$

$$\mathcal{F}_2 = \langle F_2, G_3 \cup G_4 \rangle. \quad (7)$$

Справедлива следующая теорема [4].

Теорема 3. *Пусть вероятностная мера P определяется ограниченной функцией плотности распределения, I — ЗИП типа S_d, $\mathcal{F}_2$ — базовое множество, задаваемое соотношениями* (4)–(7). *Тогда* $0 \leqslant T(I, \mathcal{F}_2) - R(I) \leqslant 2n - 1$.

Задача интервального поиска состоит в поиске в конечном подмножестве n-мерного пространства всех тех точек, которые попадают в n-мерный параллелепипед-запрос. Пусть $X_{in} = \{\tilde{x} = (u_1, v_1, \ldots, u_n, v_n)$: $0 < u_i \leqslant v_i \leqslant 1, \ i = 1, 2, \ldots, n\}$. Отношение поиска ρ_2 определено на $X_{in} \times X_n$

и задается следующим соотношением: $(u_1, v_1, \ldots, u_n, v_n)\rho_2(y_1, \ldots, y_n) \Longleftrightarrow$ $\Longleftrightarrow u_i \leqslant y_i \leqslant v_i$, $i = 1, 2, \ldots, n$. Тогда тип $S_{in} = \langle X_{in}, X_n, \rho_2, \sigma, P\rangle$ назовем типом интервального поиска. Пусть

$$G_5 = \{g^1_{i,\cdot,m}(u_1, v_1, \ldots, u_n, v_n) =]u_i \cdot m[: i \in \{1, 2, \ldots, n\},\ m = 1, 2, 3\ldots\}, \quad (8)$$

$$G_6 = \{g^2_{i,\cdot,m}(u_1, v_1, \ldots, u_n, v_n) =]v_i \cdot m[: i \in \{1, 2, \ldots, n-1\},\ m = 1, 2, 3\ldots\}, \quad (9)$$

$$G_7 = \{g^1_{i,\leqslant,a}(u_1, v_1, \ldots, u_n, v_n) = \begin{cases} 1, & \text{если } u_i \leqslant a \\ 2, & \text{если } u_i > a \end{cases} : i \in \{1, 2, \ldots, n\},\ a \in (0, 1]\}, \quad (10)$$

$$G_8 = \{g^2_{i,<,a}(u_1, v_1, \ldots, u_n, v_n) = \begin{cases} 1, & \text{если } v_i < a \\ 2, & \text{если } v_i \geqslant a \end{cases} : i \in \{1, 2, \ldots, n-1\},\ a \in (0, 1]\}. \quad (11)$$

$$G_9 = \{g_{-,m}(u_1, v_1, \ldots, u_n, v_n) = \begin{cases} 1, & \text{если } 0 \leqslant v_n - u_n < 1/m \\ 2 & \text{в противном случае} \end{cases} : m = 1, 2, 3\ldots\}, \quad (12)$$

$$F_3 = \{f_a(u_1, v_1, \ldots, u_n, v_n) = \begin{cases} 1, & \text{если } u_n \leqslant a \text{ и } v_n \geqslant a \\ 0 & \text{в противном случае} \end{cases} : a \in (0, 1]\}. \quad (13)$$

$$\mathcal{F}_3 = \langle F_3, G_5 \cup G_6 \cup G_7 \cup G_8 \cup G_9\rangle. \quad (14)$$

Справедлива следующая теорема [2].

Теорема 4. *Пусть вероятностная мера P определяется ограниченной функцией плотности распределения с ограниченными частными производными первого порядка, I — ЗИП типа S_{in}, $\mathcal{F}_3$ — базовое множество, задаваемое соотношениями* (8)–(14). *Тогда* $0 \leqslant T(I, \mathcal{F}_3) - R(I) \leqslant 4n + 1$.

Если через k обозначить мощность библиотеки, то для двумерной задачи интервального поиска объем памяти, необходимый алгоритму, на котором достигается оценка теоремы 4, равен $O(k^3)$. С целью понижения объема памяти в [6] разработана модификация алгоритма Бентли-Маурера, сохраняющая порядки времени поиска в худшем случае и объема памяти при снижении среднего времени поиска (без времени перечисления ответа) до константы. На основе этого алгоритма получена следующая оценка.

Теорема 5. *Пусть I — двумерная задача интервального поиска, вероятностная мера P определяется ограниченной функцией плотности распределения с ограниченными частными производными первого порядка, $\mathcal{F}_3$ — базовое множество, задаваемое соотношениями* (8)–(14). *Тогда для любого натурального M такого, что $1 \leqslant M \leqslant 2 \ln k$, справедливо*

$$0 \leqslant T\left(I, \mathcal{F}_3, (2/3)Mk^{1+2/M} + O(k^{1+1/M})\right) - R(I) \leqslant 14M - 4.$$

Технология проектирования физической организации баз данных, основанная на ИГМД, состоит в выделении классов однотипных вопросов к базе данных, оформляемых в виде задач информационного поиска, выделении для каждой задачи информационного поиска множества элементарных операций над запросами, оформляемого в виде базового множества, и синтезе оптимального информационного графа, решающего данную задачу информационного поиска. Этот информационный граф описывает оптимальную структуру данных, соответствующую заданным целям оптимизации (среднему времени поиска, времени поиска в худшем случае, объему памяти).

Представляется, что данное направление исследований является точкой роста теории БД.

Специальное направление в теории БД составляет защищенность данных от случайного или преднамеренного доступа к ним несанкционированных пользователей. Интенсивное развитие всемирной компьютерной сети Internet делает это направление особенно актуальным.

СПИСОК ЛИТЕРАТУРЫ

1. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. — М.: Мир, 1979.
2. Гасанов Э. Э. Мгновенно решаемые задачи поиска // Дискретная математика. — 1996. — Т. 8, вып. 3. — С. 119–134.
3. Гасанов Э. Э. Нижняя оценка сложности информационных сетей для одного отношения частичного порядка // Дискретная математика. — 1996. — Т. 8, вып. 4. — С. 108–122.
4. Гасанов Э. Э. Функционально-сетевые базы данных и сверхбыстрые алгоритмы поиска. — М.: РГГУ, 1997.
5. Гасанов Э. Э. Информационно-графовая модель хранения и поиска данных // Интеллектуальные системы. — Т. 3, № 3–4. — 1998. — С. 163–192.
6. Гасанов Э. Э., Кузнецова И. В. О функциональной сложности двумерной задачи интервального поиска // Дискретная математика. — 2002. — Т. 14, вып. 1.
7. Гасанов Э. Э., Луговская Ю. П. Константный в худшем случае алгоритм поиска идентичных объектов // Дискретная математика. — 1999. — Т. 11, вып. 4. — С. 139–144.
8. Ершов А. П. О программировании арифметических операторов // Докл. АН СССР. — Т. 118. — С. 427–430.
9. Мейер Д. Теория реляционных баз данных. — М.: Мир, 1987.
10. Препарата Ф., Шеймос М. Вычислительная геометрия: Введение. — М.: Мир, 1989.
11. Ульман Дж. Основы систем баз данных. — М.: Финансы и статистика, 1983.
12. Язык описания данных КОДАСИЛ. — М.: Статистика, 1981.
13. Atkinson M., Bancilhon F., DeWitt D., Dittrich K., Maier D., Zdonic S. The Object-Oriented Database System Manifesto // Proc. 1st DOOD. — Kyoto, 1989.
14. Bentley J. L. Multidimensional binary search trees used for associative searching // Commun. Ass. Comput. Mach. (Sept. 1975). — V. 18. — P. 509–517.
15. Bentley J. L. Decomposable searching problems // Info. Proc. Lett. — 1979. — V. 8. — P. 244–251.
16. Bentley J. L., Maurer H. A. Efficient worst-case data structures for range searching // Acta Informatica. — 1980. — V. 13. — P. 155–168.
17. Bentley J. L., Shamos M. I. A problem in multivariate statistics: Algorithms, data structure and applications // Proc. 15th Allerton Conf. Commun., Contr., Comput. — 1977. — P. 193–201.
18. Bolour A. Optimal retrieval algorithms for small region queries // SIAM J. Comput. — 1981. — V 10. — P. 721–741.
19. Borodin A. B. Computational complexity — Theory and practice // Currents in Theory of Computings. — Englewood Cliffs, NJ: Printice-Hall, 1973. — P. 35–89.
20. Chazelle B. M. Filtering search: a new approach to query-answering // Proc. 24th IEEE Annu. Symp. Found. Comput. Sci. (Nov. 1983). — P. 122–132.
21. Foundations of deductive databases and logic programming // Minker J. (ed.). — Los Altos: Morgan Kaufman, 1988.
22. Information Management System Virtual Storage (IMS/VS), General Information Manual GH20-1260. — IBM, White Plains, New York, 1974.
23. Lee D. T., Wong C. K. Worst case analysis for region and partial region searches in multidimensional binary search trees and balansed quad trees // Acta Informatica. — 1977. — V. 9. — P. 23–29.

24. Lueker G. S. A data structure for ortogonal range queries // Processing of the 19th Annual IEEE Symposium on Foundations of Computer Science. — 1978. — P. 28–34.
25. Thalheim B. Entity-Relationship Modeling. Foundations of Database Technology. — Berlin: Springer, 2000.
26. Willard D. E. Predicate-oriented database search algorithms. — Ph. D. dissertation. — Harvard Univ., Cambridge, MA, 1978.

Поступило в редакцию 6 V 2001