О сложности хранения и поиска информации*

Э. Э. Гасанов

Построена новая общая модель хранения и поиска информации, называемая информационно-графовой, частными случаями которой являются известные модели представления данных. Изучены основные свойства этой модели и решена проблема оптимального синтеза информационных графов для широкого класса задач поиска, включающего наиболее часто используемые на практике задачи поиска в базах данных.

1. Введение

В последние десятилетия активно развивается новое научное направление, связанное с оптимальным хранением и поиском информации, именуемое теорией информационного поиска. Одним из главных носителей этого направления является теория баз данных. Возникшее под влиянием практических задач, оно и сейчас в основном обслуживает приложения, а собственно теоретическая его часть, как представляется, обретает контуры. Как всякая научная дисциплина это направление должно характеризоваться следующими чертами: предметом исследования, проблематикой, методами и результатами. В развитой теории каждая из этих черт должна иметь достаточно общий характер. В то же время важно отметить, что молодые дисциплины возникают, как правило, через рассмотрение отдельных конкретных важных примеров, которые затем с развитием дисциплин

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (грант 01–01–00748).

обобщаются как в постановочной, так и в проблемно-методологической частях. Подобных примеров достаточно много в кибернетике, информатике и других разделах науки. К числу характерных из них может быть отнесена теория управляющих систем. В своей практической деятельности человек столкнулся с конкретными видами таких систем, которые далее играли роль модельных управляющих систем. В инженерном деле — это вентильные и контактные схемы, схемы из функциональных элементов и некоторые другие, в математике — формулы, алгоритмы и т. д., в биологии — нейроны, нейронные сети, автоматы и т. п. Для этих видов управляющих систем рассматривались две такие главные задачи анализа и синтеза соответствующих управляющих систем [1, 2]. Первая состояла в изучении «поведения» таких систем, а вторая — в создании соответствующей системы с заданным «поведением». На первом этапе эти постановки связывались непосредственно с модельными системами и для каждой из них разрабатывался конкретный метод решения. Со временем наступил этап, когда большинство из этих систем могли уже рассматриваться с единых позиций, и исследование указанных общих задач достигалось уже с помощью общих методов решения [3]. Хотя по-прежнему модельные управляющие системы, имея свою специфику, продолжают оставаться в центре внимания теории управляющих систем.

Аналогичный путь развития проходит теория информационного поиска. В ней также первоначально возникли конкретные примеры способов хранения и представления данных и соответствующих этим способам алгоритмов поиска информации. К их числу относятся лексикографические, древовидные, реляционные и др. Задачи поиска для них имеют конкретные виды и модификации такие, как задача поиска идентичных объектов, задача о близости, включающий поиск и др. Они играют роль модельных задач для выбранных способов хранения информации и изучались на протяжении многих лет, привлекая каждый раз для своего решения специальные исследовательские средства, которые носили ограниченный по своим возможностям характер.

Тем самым, можно считать, что современное состояние теории информационного поиска напоминает то состояние теории управляющих систем, которое соответствовало первому этапу развития послед-

ней, когда накапливались данные лишь о конкретных видах модельных управляющих систем. В то же время, как выяснилось, опыт развития теории управляющих систем в своей методологической части давал возможность сделать попытку с более общих позиций провести исследование как модельных баз данных, так и модельных задач для них, с соответствующей разработкой достаточно общей теории.

Мы предлагаем новую модель данных (частными случаями которой могут считаться уже известные) с наследственно определенными средствами поиска информации, с соответствующими понятиями сложности такого поиска, а также разрабатываем основы теории решения базовых задач поиска применительно к этой модели. Если продолжить аналогию с теорией синтеза управляющих систем, то можно отметить, что различным видам управляющих систем соответствуют различные виды хранения и представления данных (модели данных), классам функций, исследуемым в теории синтеза, соответствуют типы задач поиска, исследуемые в теории информационного поиска. И в теории синтеза и в теории поиска вводятся понятия сложности и ставятся задача оптимального синтеза и задача исследования функций сложности шенноновского типа. Таким образом, мы стремимся к тому, чтобы приблизить состояние теории информационного поиска по степени продвинутости к современному состоянию теории управляющих систем.

Уточним сказанное. Во-первых, мы предлагаем новую формализацию понятия задачи поиска. Тип задач поиска охватывает класс однотипных вопросов к базе данных. Тип задач поиска заключает в своем определении три объекта: множество запросов, множество записей и бинарное отношение, заданное на декартовом произведении этих множеств, называемое отношением поиска. Здесь запись — это поисковый образ элемента данных, то есть поле или множество полей элемента данных, которые представляют интерес в данном типе вопросов. Запрос — это минимальный элемент, содержащий суть вопроса. Запрос совместно с отношением очерчивает тот круг объектов, которые отвечают на данный вопрос. Задача поиска заданного типа получается выделением из множества записей конечного подмножества, называемого библиотекой. А именно, задача поиска состоит в том, чтобы по произвольному запросу перечислить все записи из биб-

лиотеки, находящиеся в заданном отношении с запросом (удовлетворяющие запросу). При фиксации отношения поиска каждая запись задает предикат, определенный на множестве запросов, который равен 1, если данная запись удовлетворяет запросу — аргументу функции. Поэтому если вернуться к аналогии с теорией синтеза управляющих систем, то тип задач поиска есть способ описания некоторого конкретного класса предикатов, задаваемых на множестве запросов, а задача поиска — это конкретное подмножество предикатов из этого класса.

Во-вторых, мы предлагаем новую управляющую систему, называемую информационным графом, которая в общей иерархии теории управляющих систем находится в не очень высоких слоях залегания и является в некотором смысле обобщением контактных схем. Фактически нам нужны лишь графы, дискретные функции и вычисление волновых процессов на графах, и этого хватает, чтобы с достаточно общих позиций посмотреть на ту разрозненную картину, которая наблюдается в теории информационного поиска.

В предлагаемой информационно-графовой модели данных структура данных задается ориентированным графом (называемым информационным), ребра и вершины которого нагружены элементами данных и функциями, определенными на множестве запросов. В графе выделена одна вершина, называемая корнем и ассоциируемая со входом, а вершины графа, нагруженные элементами данных, ассоциируются с выходами. Этот же граф описывает алгоритм поиска, на вход которого поступает запрос, а на выходе получается некоторое подмножество данных. При этом процесс поиска начинается с корня и распространяется в зависимости от значений нагрузочных функций на запросе, возможно, сразу по нескольким направлениям. Если этот волновой процесс на графе достигает элементов данных, то эти элементы включаются в ответ алгоритма на исходный запрос. Информационный граф будет решать некоторую задачу поиска, если для произвольного запроса ответ на этот запрос содержит все те и только те записи из библиотеки, которые удовлетворяют запросу. Таким образом, информационный граф с одной стороны дает новую концепцию хранения данных, а с другой стороны предлагает новый подход к поиску информации как волнового процесса на графах, управляе-

мого нагрузочными функциями. Нагрузочные функции, которые называются базовыми, разделены на два класса — предикаты и переключатели, и являются одним из основных управляющих параметров модели. Нагрузочные функции по сути определяют функции проводимости между вершинами графа, и проблема нахождения решения задачи поиска сводится к проблеме синтеза информационного графа, реализующего систему функций, задаваемую задачей поиска.

Так же, как логическая сеть со свободными элементами [4] обобщает известные в теории управляющих систем виды управляющих систем, так и информационно-графовая модель обобщает наиболее известные модели данных. Понятно, что алгоритмы и конструкции, используемые в *древовидных базах данных*, описываются древовидными информационными графами. *Сетевые базы данных*, естественным образом перекладываются на язык информационно-графовой модели, при этом ясно, что со структурной точки зрения они по существу будут представляться графами. В *дедуктивных базах* нужные данные и знания получаются путем логического вывода, поэтому алгоритм поиска, используемый в дедуктивных базах данных, при переходе на язык информационных графов приводит к константному дереву, который отражает суть дерева логического вывода. В *реляционных базах* данные представляются в виде таблиц, при этом алгоритм поиска, ассоциируемый с таким представлением данных, есть алгоритм перебора, который естественно легко описывается древовидным информационным графом специального вида.

Информационные графы позволяют ввести новое понятие сложности поиска. Это понятие новое как с точки зрения теории управляющих систем, так и с точки зрения теории баз данных. В теории управляющих систем обычно под сложностью понимается или число ребер, или число элементов-функций, а здесь сложность понимается как часть графа, захваченного волновым процессом, и существенно зависит от значений нагрузочных функций, и тем самым не является просто количественной характеристикой графа такой, как число ребер или вершин. Новизна же в теории баз данных заключается в том, что такое введение сложности после осреднения по множеству запросов адекватно соответствует среднему времени поиска — традиционно трудной для изучения характеристики алгоритмов поиска

информации. Кроме того, при соответствующем введении сложности информационные графы оказываются удобными для изучения как параллельных, так и фоновых алгоритмов поиска. И, наконец, в информационных графах совсем просто контролируется такой важный управляющий параметр в задачах информационного поиска, как объем памяти, который в данном случае характеризуется количеством ребер графа.

Рассматривается несколько модельных типов задач поиска, являющихся наиболее распространенными задачами поиска в базах данных. Выбор модельных типов определяется как повсеместностью использования их в базах данных, так и частотой цитирования в литературе [5–22]. Эти модельные типы можно разбить их на 3 крупных базовых класса. Первый класс включает в себя задачи поиска, в которых для почти всех запросов ответ на них содержит ограниченное малой константой число элементов. Этот класс получил название задач поиска с коротким ответом. Представителем этого класса является задача поиска идентичных объектов.

Второй класс, названный задачами поиска на частично-упорядоченных множествах данных, состоит из задач, в которых в ответ на запрос надо перечислить все элементы базы данных, которые в заданном частичном порядке меньше чем запрос. Представителями этого класса являются задача включающего поиска и задача о доминировании.

И, наконец, третий класс содержит так называемые задачи интервального поиска, результат которых в некотором смысле можно рассматривать как пересечение решений двух задач из второго класса.

Для базовых задач поиска ставится и решается проблема оптимального синтеза, которая состоит в построении для заданной задачи информационного поиска информационного графа, который решает эту задачу и имеет наименьшую или близкую к ней сложность.

Полученный свод результатов, описывающих оптимальное решение базовых классов, назовем каноническим эффектом, и мы хотим понять, насколько чувствительна основная модель по отношению к каноническому эффекту при вариации 3-х основных управляющих параметров модели, таких как объем памяти, имеющийся в распоряжении (то есть число ребер информационного графа), множество

функций, которые разрешается использовать при решении (то есть множество базовых функций, используемых при нагрузке графа), и ε-расширение запроса. Показывается, что при любой вариации, кроме ε-расширения запроса при достаточно малых ε, мы уходим от канонического эффекта.

Для решения задач оптимального синтеза для базовых классов разработаны следующие 3 основных метода.

Первый метод мы называем методом оптимальной декомпозиции. Он состоит в таком разбиении задачи на подзадачи, которые допускают простое решение и при этом сложность поиска подзадачи, дающей решение исходной задачи, также осуществляется просто. Этот метод использовался при решении опорных или одномерных задач поиска.

Второй метод, называемый методом снижения размерности, применяемый к многомерным задачам, сводится к тому, чтобы с помощью некоторых опорных задач последовательно понижать размерность задачи и в конце концов свести ее к опорной задаче, решение которой уже известно.

Третий метод назван методом характеристических носителей графа и использовался при получении нижних оценок. Он заключается в выделении в информационном графе, являющемся оптимальным решением, подграфов с заданными свойствами (характеристических носителей) и в последующем подсчете сложности характеристических носителей.

Данная работа содержит обзор результатов автора, полученных им в этом направлении.

Автор выражает глубокую благодарность академику В. Б. Кудрявцеву и профессору А. С. Подколзину за внимание и помощь в работе.

2. Информационно-графовая модель данных

В задачах поиска, возникающих в базах данных, имеется 3 основных объекта:

- *множество запросов* X с заданным на нем вероятностным пространством;

- множество потенциальных ответов Y, будем называть элементы этого множества *записями*;
- бинарное отношение ρ, заданное на $X \times Y$, называющееся отношением поиска и описывающее критерий семантического соответствия записи запросу, то есть если $x\rho y$, то будем говорить, что запись y удовлетворяет запросу x;

В достаточно общем случае значительный интерес представляет описываемая ниже проблема, которую мы назовем задачей информационного поиска. Тройку $\langle X, Y, \rho\rangle$ будем называть *типом задач информационного поиска*, а тройку $\langle X, V, \rho\rangle$ (или четверку $\langle X, V, \rho; Y\rangle$), где V — конечное подмножество Y, называемое *библиотекой*, — *задачей информационного поиска* (ЗИП). Содержательно будем считать, что ЗИП $I = \langle X, V, \rho; Y\rangle$ состоит в перечислении для произвольно взятого запроса $x \in X$ всех тех и только тех записей из V, которые находятся в отношении ρ с запросом x, то есть удовлетворяют запросу x.

Реально эта проблема допускает вариацию как за счет уточнения самой задачи, так и за счет допущения разных предположений относительно базовых компонент X, Y, ρ, V, составляющих ЗИП.

Опишем основной объект, который называется информационным графом (ИГ). Вводить ИГ мы будем, одновременно иллюстрируя его на примере одномерной задачи интервального поиска, которая состоит в поиске в конечном подмножестве отрезка $[0, 1]$ вещественной прямой всех тех точек, которые попадают в отрезок-запрос.

Сначала задаются 4 множества:

- множество запросов X;
- множество записей Y;
- множество F одноместных предикатов, заданных на множестве X;
- множество G одноместных переключателей, заданных на множестве X (переключатели — это функции, область значений которых является начальным отрезком натурального ряда).

В примере эти множества имеют вид:

- $X_{int1} = \{(u, v) : 0 < u \leqslant v \leqslant 1\}$;
- $Y_{int1} = (0, 1]$;
- $F = F_1 \bigcup F_2$, где $F_1 = \{f^1_{\leqslant,a} : a \in (0, 1]\}$, $F_2 = \{f^2_{\geqslant,a} : a \in (0, 1]\}$,

$$f^1_{\leqslant,a}(u, v) = \begin{cases} 1, & \text{если } u \leqslant a \\ 0, & \text{если } u > a \end{cases},$$

$$f^2_{\geqslant,a}(u, v) = \begin{cases} 1, & \text{если } v \geqslant a \\ 0, & \text{если } v < a \end{cases},$$

- $G = G_1 \bigcup G_2 \bigcup G_3$, где $G_1 = \{g_{*,m} : m \in \mathbf{N}\}$, $G_2 = \{g_{-,m} : m \in \mathbf{N}\}$, $G_3 = \{g_{\leqslant,a} : a \in (0, 1]\}$, $g_{*,m}(u, v) =]u \cdot m[$,

$$g_{-,m}(u, v) = \begin{cases} 1, & \text{если } v - u < 1/m \\ 2, & \text{если } v - u \geqslant 1/m \end{cases},$$

$$g_{\leqslant,a}(u, v) = \begin{cases} 1, & \text{если } u \leqslant a \\ 2, & \text{если } u > a \end{cases}.$$

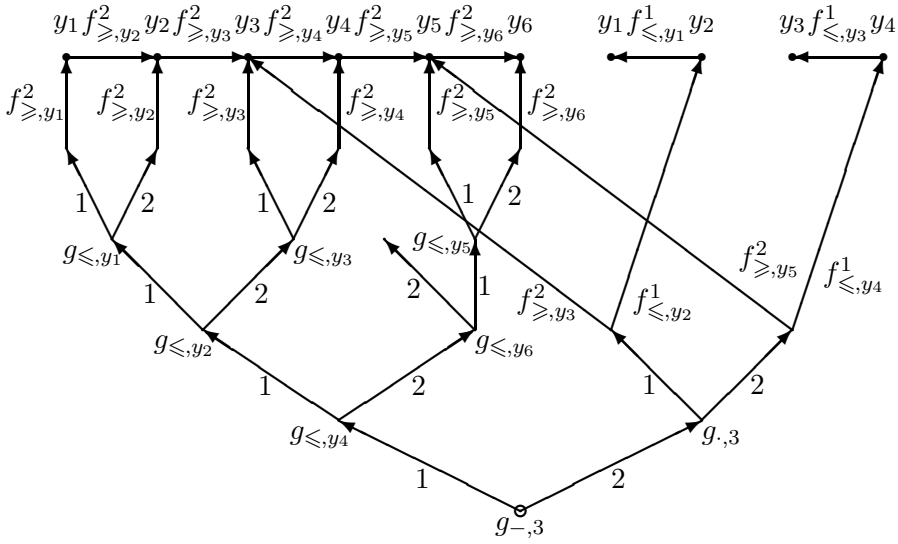

Рис. 1. Решение одномерной задачи интервального поиска.

ИГ определяется следующим образом. Берется конечная многополюсная ориентированная сеть. В ней выбирается некоторый полюс, который называется корнем. На рисунке 1 он изображен полым

кружком. Остальные полюса называются листьями (на рисунке они изображены жирными точками) и им приписываются записи из Y (на рисунке это символы y с индексами), причем разным листьям могут быть приписаны одинаковые записи. Некоторые вершины сети (в том числе это могут быть и полюса) называются переключательными и им приписываются переключатели из G (на рисунке таких вершин 8). Ребра, исходящие из каждой из переключательных вершин, нумеруются начиная с 1 и называются переключательными ребрами (на рисунке таких ребер 16). Ребра, не являющиеся переключательными, называются предикатными и им приписываются предикаты из множества F (на рисунке таких ребер 17). Таким образом нагруженную многополюсную ориентированную сеть называем ИГ над базовым множеством $\mathcal{F} = \langle F, G \rangle$.

Функционирование ИГ определяется следующим образом. Скажем, что предикатное ребро проводит запрос $x \in X$, если предикат, приписанный этому ребру, принимает значение 1 на запросе x. Скажем, что переключательное ребро, которому приписан номер n, проводит запрос $x \in X$, если переключатель, приписанный началу этого ребра, принимает значение n на запросе x. Скажем, что ориентированная цепочка ребер проводит запрос $x \in X$, если каждое ребро цепочки проводит запрос x. Скажем, что запрос $x \in X$ проходит в вершину β ИГ, если существует ориентированная цепочка, ведущая из корня в вершину β, которая проводит запрос x. Скажем, что запись y, приписанная листу α, попадает в ответ ИГ на запрос $x \in X$, если запрос x проходит в лист α. Ответом ИГ U на запрос x назовем множество записей, попавших в ответ ИГ на запрос x, и обозначим его $\mathcal{J}_U(x)$. Эту функцию $\mathcal{J}_U(x)$ будем считать результатом функционирования ИГ U.

Из определения функционирования ИГ естественным образом вытекает, что каждому ИГ U можно сопоставить некую процедуру поиска.

Предполагается, что эта процедура хранит в своей (внешней) памяти структуру ИГ U. Входными данными процедуры является запрос. Выходными данными является множество записей.

Опишем эту процедуру.

Пусть на вход процедуры поступил запрос x. Вводим понятие активного множества вершин и вносим в него в начальный момент корень ИГ U и помечаем его. Далее по очереди просматриваем вершины из активного множества и для каждой из них проделываем следующее:

- если рассматриваемая вершина — лист, то запись, приписанную вершине, включаем в ответ;
- если рассматриваемая вершина переключательная, то вычисляем на запросе x переключатель, соответствующий данной вершине, и если конец ребра, исходящего из рассматриваемой вершины, нагрузка которого равна значению переключателя, непомеченная вершина, то помечаем его и включаем в множество активных вершин;
- если рассматриваемая вершина предикатная, то просматриваем по очереди исходящие из нее ребра и вычисляем значения предикатов, приписанных этим ребрам, на запросе x. Концы ребер, которым соответствуют предикаты со значениями, равными 1, если они непомеченные, помечаем и включаем в множество активных вершин;
- исключаем рассматриваемую вершину из активного множества.

Процедура завершается по исчерпании активного множества.

Таким образом, ИГ как управляющая система может рассматриваться в качестве модели алгоритма поиска, работающего над данными, организованными в структуру, определяемую структурой ИГ.

Пусть нам дана ЗИП $I = \langle X, V, \rho \rangle$.

Скажем, что ИГ U *разрешает* ЗИП $I = \langle X, V, \rho \rangle$, если для любого запроса $x \in X$ ответ на этот запрос содержит все те и только те записи из V, которые удовлетворяют запросу x, то есть

$$\mathcal{J}_U(x) = \{y \in V : x \rho y\}.$$

Если ρ_{int1} — бинарное отношение на $X_{int1} \times Y_{int1}$ такое, что

$$(u, v) \rho_{int1} y \Longleftrightarrow u \leqslant y \leqslant v,$$

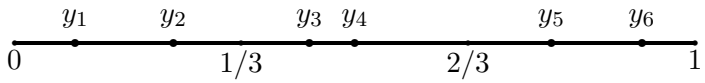

Рис. 2.

то ИГ, изображенный на рисунке 1, разрешает ЗИП $I = \langle X_{int1}, V, \rho_{int1} \rangle$, где $V = \{y_1, y_2, y_3, y_4, y_5, y_6\}$ — библиотека, изображенная на рисунке 2, причем данный ИГ, соответствует асимптотически оптимальному решению, полученному по методу оптимальной декомпозиции, описание которого применительно к данной задаче мы приведем позже.

Введем вспомогательные обозначения.

Если f — одноместный предикат, определенный на X, то множество $N_f = \{x \in X : f(x) = 1\}$ назовем *характеристическим множеством* предиката f.

Множество $O(y, \rho) = \{x \in X : x \rho y\}$ назовем *тенью* записи $y \in Y$.

Введем понятие сложности ИГ.

Пусть β — некоторая вершина ИГ. Предикат, определенный на множестве запросов, который принимает значение 1 на запросе x, если запрос проходит в вершину β, и 0 — в противном случае, назовем функцией фильтра вершины β и обозначим $\varphi_\beta(x)$.

Определим понятие сложности ИГ на запросе.

Будем считать, что время вычисления любого переключателя из G и любого предиката из F одинаково и равно 1.

Пусть нам дан некий ИГ U и произвольно взятый запрос $x \in X$. Пусть A — определенная ранее процедура, сопоставленная ИГ U.

Сложностью ИГ U на запросе x назовем число $T(U, x)$, равное количеству переключателей и предикатов, вычисленных процедурой A при подаче на его вход запроса x, то есть

$$T(U, x) = \sum_{\beta \in \mathcal{P}} \varphi_\beta(x) + \sum_{\beta \in \mathcal{R} \setminus \mathcal{P}} \psi_\beta \cdot \varphi_\beta(x),$$

где $\mathcal{R}$ — множество вершин ИГ U, $\mathcal{P}$ — множество переключательных вершин ИГ U, ψ_β — количество ребер, исходящих из вершины β.

Величина $T(U,x)$ характеризует время работы процедуры A при подаче на его вход запроса x.

Введем понятие сложности ИГ как среднее значение сложности ИГ на запросе, взятое по множеству всех запросов. С этой целью введем *вероятностное пространство* над множеством запросов X, под которым будем понимать тройку $\langle X, \sigma, \mathbf{P}\rangle$, где σ — некоторая *алгебра подмножеств* множества X, $\mathbf{P}$ — *вероятностная мера* на σ, то есть аддитивная мера, такая, что $\mathbf{P}(X) = 1$.

Скажем, что базовое множество $\mathcal{F}$ *измеримое*, если каждая функция из $\mathcal{F}$ — измеримая (относительно алгебры σ). Далее всюду будем предполагать, что базовое множество измеримое. В этом случае для любого ИГ U над $\mathcal{F}$ функция $T(U,x)$ как функция от x измерима.

Сложностью ИГ U назовем математическое ожидание величины $T(U,x)$, то есть число

$$T(U) = \mathbf{M}_x\, T(U,x).$$

Объемом $Q(U)$ ИГ U назовем число ребер в ИГ U.

Пусть нам дана некая ЗИП I. *Сложностью задачи* I *при базовом множестве* $\mathcal{F}$ *и заданном объеме* q назовем число

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

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

Число

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

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

Если $I = \langle X, V, \rho\rangle$, то величина $R(I) = \sum_{y\in V} \mathbf{P}(O(y,\rho))$ есть средняя длина ответа ЗИП I.

Справедлива следующая теорема [23].

Теорема 1 (мощностная нижняя оценка). *Если* $I = \langle X, V, \rho\rangle$ *— произвольная ЗИП,* $\mathcal{F}$ *— измеримое базовое множество, такое, что множество* $\mathcal{U}(I,\mathcal{F}) \neq \varnothing$, *то* $T(I,\mathcal{F}) \geqslant R(I)$.

Этот результат был получен с помощью метода характеристических носителей графа.

3. Решение проблемы оптимального синтеза для базовых задач

Если существует такой ИГ $U \in \mathcal{U}(I, \mathcal{F})$, что $T(U) = T(I, \mathcal{F})$, то ИГ U будем называть *оптимальным* для ЗИП I.

Для модельных классов ставится проблема синтеза оптимального ИГ.

Среди **задач поиска, в которых вероятность появления в ответе более** c **записей** ($c = \text{const}$) **равна нулю**, наиболее подробно исследована ситуация, когда $c = 1$.

Для таких задач показано, что оптимальные ИГ древовидны, а в случае, когда тени всех записей библиотеки имеют равную вероятность (такие ЗИП названы обладающими G-свойством) справедлив следующий результат [24].

Теорема 2. *Если* $I = \langle X, V, \rho \rangle$ *— ЗИП, обладающая* G*-свойством,* $\mathcal{F} = \langle F, \varnothing \rangle$ *— некоторое специальное базовое множество, то*

$$\mathbf{P}(O(y, \rho)) \cdot R(k) \leqslant T(I, \mathcal{F}) \leqslant \mathbf{P}(O(y, \rho)) \cdot R(k) + 1,$$

где $y \in V$, $k = |V|$ *— мощность библиотеки* V*,*

$$R(k) = 3k[\log_3 k] + 4(k - 3^{[\log_3 k]}) + \max(0, k - 2 \cdot 3^{[\log_3 k]}).$$

Здесь и далее формулировки теорем носят несколько упрощенный характер и служат только для того, чтобы отразить общую картину. Строгие формулировки можно найти по ссылкам. Этот результат был получен методом характеристических носителей графа.

Задача **поиска идентичных объектов** состоит в поиске в множестве объекта, идентичного объекту-запросу, и формально принадлежит типу $S_{id} = \langle (0, 1], (0, 1], =, \sigma, \mathbf{P} \rangle$. **Задача о близости** состоит в поиске в линейно-упорядоченном множестве объекта, ближайшего к объекту-запросу, и принадлежит типу $S_{ne} = \langle (0, 1], (0, 1], \rho_{ne}, \sigma, \mathbf{P} \rangle$, 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

$$F_3 = \{f_{=,a}(x) = \begin{cases} 0, & \text{если } x \neq a \\ 1, & \text{если } x = a \end{cases} : a \in (0,1]\}, \tag{1}$$

$$G_4 = \{g_{\leqslant,a}(x) = \begin{cases} 1, & \text{если } x \leqslant a \\ 2 & \text{в противном случае} \end{cases} : a \in (0,1]\}, \tag{2}$$

$$G_5 = \{]x \cdot m[: m = 1,2,3\ldots\},\ \mathcal{F} = \langle F_3, G_4 \cup G_5\rangle. \tag{3}$$

Справедлива теорема [25].

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

Эта теорема получена методом оптимальной декомпозиции.

В [26] для задачи поиска идентичных объектов приводится алгоритм, который в типичной ситуации при затратах памяти k^2 обеспечивает время поиска, в худшем случае равное 2.

ЗИП с отношением поиска, являющимся отношением линейного предпорядка — первая из задач, относящихся ко второму классу задач поиска на частично-упорядоченных множествах данных.

Отношение линейного предпорядка — это отношение, удовлетворяющее условиям рефлексивности, транзитивности и связности.

Будем рассмаривать следующий тип: $S_{lin} = \langle X, X, \overset{l}{\succeq}\rangle$, где X — некоторое множество, $\overset{l}{\succeq}$ — некоторое отношение линейного предпорядка на $X \times X$.

Пусть $\mathcal{K} = \{\chi_{a,\overset{l}{\succeq}}(x) : a \in X\}$. Справедлива следующая теорема [27].

Теорема 4. *Для любой ЗИП* $I = \langle X, V, \overset{l}{\succeq}\rangle$ *типа* S_{lin} *существует оптимальный ИГ над базовым множеством* $\mathcal{F} = \langle \mathcal{K}, \varnothing\rangle$ *и*

$$T(I, \mathcal{F}) = 1 + R(I) - \min_{y \in V} \mathbf{P}(O(y, \overset{l}{\succeq})).$$

Для ЗИП типа S_{lin} исследовалось также параллельное решение [28], которое предполагает, что ИГ обрабатывается сразу несколькими вычислителями, при этом выделяется два подхода: когда ИГ распределяется на части между вычислителями и каждый вычислитель обрабатывает только свою часть (сепаративный подход); когда вычислители совместно обрабатывают ИГ (кооперативный подход). Получено оптимальное параллельное решение в случае сепаративного подхода, и показано существование таких ЗИП типа S_{lin}, для которых кооперативный подход дает лучшие результаты, чем сепаративный подход.

Задача **включающего поиска** принадлежит следующему типу: $S_{bool} = \langle B^n, B^n, \overset{b}{\succeq}\rangle$, где B^n — *единичный n-мерный куб*, $\overset{b}{\succeq}$ — отношение поиска на $B^n \times B^n$, определяемое следующим соотношением

$$(x_1, \ldots, x_n)\overset{b}{\succeq}(y_1, \ldots, y_n) \Longleftrightarrow x_i \geqslant y_i,\ i = \overline{1,n},$$

причем на B^n задана равномерная вероятностная мера, то есть для $\forall x \in B^n \quad \mathbf{P}(x) = 1/2^n$ и $\forall A \subseteq B^n \quad \mathbf{P}(A) = |A|/2^n$. Справедлива следующая теорема [29].

Теорема 5. *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 $I = \langle B^n, V, \overset{b}{\succeq}\rangle$ типа S_{bool} $T(I, \mathcal{F}) \geqslant 2R(I)$ и существуют такие ЗИП $I = \langle B^n, V, \overset{b}{\succeq}\rangle$ типа S_{bool}, что $T(I, \mathcal{F}) = 2R(I)(1 + \bar{o}(1))$ при $n \to \infty$.*

Нижняя оценка этой теоремы была получена с помощью метода характеристических носителей графа. Приведем краткое описание этого метода применительно к задаче включающего поиска. На первом этапе показывается, что для каждой записи из библиотеки задачи в ИГ, решающем данную задачу, существует так называемая главная цепь, то есть цепочка ребер, ведущая из корня ИГ в лист, которому приписана данная запись, и по этой цепочке проходят все запросы, которым удовлетворяет данная запись. Далее перебирая различные

варианты пересечения главных цепей, показывается, что библиотеку можно разбить на непересекающиеся части таким образом, что каждой части можно сопоставить свое подмножество ребер графа (такие подмножества обычно имеют вид метелки), суммарная сложность которых не меньше, чем удвоенная сумма вероятностей теней записей из данной части.

Как видно теорема 5 дает асимптотику функции Шеннона. Кроме того для включающего поиска была получена асимптотика логарифма сложности для почти всех задач и для средней сложности по задачам.

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 $X_{dom} = (0,1]^n$. Отношение поиска ρ_{dom} определено на $X_{dom} \times X_{dom}$ и задается следующим соотношением $(x_1, x_2, \ldots, x_n)\rho_{dom}(y_1, y_2, \ldots, y_n) \Longleftrightarrow y_i \leqslant x_i,\ i = 1, 2, \ldots, n$. Тогда тип $S_{dom} = \langle X_{dom}, X_{dom}, \rho_{dom}, \sigma, \mathbf{P}\rangle$ назовем типом задачи о доминировании. Пусть

$$G_6 = \{g_{i,\cdot,m}(x_1, \ldots, x_n) = \\ =]x_i \cdot m[: i \in \{1, 2, \ldots, n-1\},\ m = 1, 2, 3 \ldots\}, \tag{4}$$

$$G_7 = \{g_{i,<,a}(x_1, \ldots, x_n) = \\ = \left\{ \begin{array}{l} 1, \text{ если } x_i < a \\ 2, \text{ если } x_i \geqslant a \end{array} \right. : i \in \{1, 2, \ldots, n-1\},\ a \in (0, 1]\}, \tag{5}$$

$$F_4 = \{g_{n,\geqslant,a}(x_1, \ldots, x_n) = \left\{ \begin{array}{l} 0, \text{ если } x_n < a \\ 1, \text{ если } x_n \geqslant a \end{array} \right. : a \in (0, 1]\}, \tag{6}$$

$$\mathcal{F} = \langle F_4, G_6 \cup G_7\rangle. \tag{7}$$

Справедлива следующая теорема [30].

Теорема 6. *Пусть вероятностная мера* $\mathbf{P}$ *определяется ограниченной функцией плотности распределения,* I *— ЗИП типа* S_{dom}*,* $\mathcal{F}$ *— базовое множество, задаваемое соотношениями (4)–(7). Тогда, если функция плотности вероятности* $p(x) \leqslant c$*, то*

$$0 < T(I, \mathcal{F}, \binom{k+n-1}{n} + (3+c) \cdot \sum_{i=1}^{n-1} \binom{k+i-1}{i}) - R(I) \leqslant 2\,n - 1.$$

Этот результат был получен с помощью метода снижения размерности. Приведем краткое описание этого метода применительно к n-мерной задаче о доминировании. Возьмем произвольный запрос. Он описывает n требований к ответу: по каждой из n компонент элементы ответа не должны превышать соответствующую компоненту запроса. С помощью решения задачи о близости (опорная задача, оптимальное решение которой приводится в теореме 3) мы получаем подмножество библиотеки, состоящее из всех записей, удовлетворяющих одному из n требований. Далее опять применяем к полученному подмножеству библиотеки задачу о близости и еще раз снижаем размерность. Таким образом за $n-1$ применений задачи о близости (то есть в среднем за $2(n-1)$ вычислений) мы придем к одномерной задаче о доминировании, оптимальное решение которой приводится в теореме 4.

Для двумерной задачи о доминировании исследовалось также решение задачи в фоновом режиме [31]. Для алгоритмов поиска в фоновом режиме предполагается наличие внешнего объекта, называемого пользователем. Элементы ответа на запрос при этом считаются поступающими по мере нахождения, каждый элемент ответа обрабатывается пользователем в течении некоторого времени, а сложность алгоритма определяется как время простоя пользователя. Найдено фоновое решение двумерной задачи о доминировании, которое в типичной ситуации при линейных затратах памяти имеет константную временную сложность.

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

$$(u_1, v_1, \ldots, u_n, v_n)\rho_{intn}(y_1, \ldots, y_n) \Longleftrightarrow u_i \leqslant y_i \leqslant v_i,\ i = 1, 2, \ldots, n.$$

Тогда тип $S_{intn} = \langle X_{intn}, Y_{intn}, \rho_{intn}, \sigma, \mathbf{P}\rangle$ назовем типом интервального поиска. Пусть

$$G_8 = \{g^1_{i,\cdot,m}(u_1, v_1, \ldots, u_n, v_n) = \\ =]u_i \cdot m[: i \in \{1, 2, \ldots, n\},\ m = 1, 2, 3 \ldots\}, \tag{8}$$

$$G_9 = \{g^2_{i,\cdot,m}(u_1, v_1, \ldots, u_n, v_n) = \\ =]v_i \cdot m[: i \in \{1, 2, \ldots, n-1\},\ m = 1, 2, 3 \ldots\}, \tag{9}$$

$$G_{10} = \{g^1_{i,\leqslant,a}(u_1, v_1, \ldots, u_n, v_n) = \\ = \left\{ \begin{array}{l} 1,\ \text{если } u_i \leqslant a \\ 2,\ \text{если } u_i > a \end{array} \right. : i \in \{1, 2, \ldots, n\},\ a \in (0, 1]\}, \tag{10}$$

$$G_{11} = \{g^2_{i,<,a}(u_1, v_1, \ldots, u_n, v_n) = \\ = \left\{ \begin{array}{l} 1,\ \text{если } v_i < a \\ 2,\ \text{если } v_i \geqslant a \end{array} \right. : i \in \{1, 2, \ldots, n-1\},\ a \in (0, 1]\}. \tag{11}$$

$$G_{12} = \{g_{-,m}(u_1, v_1, \ldots, u_n, v_n) = \\ = \left\{ \begin{array}{ll} 1, & \text{если } 0 \leqslant v_n - u_n < 1/m \\ 2 & \text{в противном случае} \end{array} \right. : m = 1, 2, 3 \ldots\}, \tag{12}$$

$$F_5 = \{f_a(u_1, v_1, \ldots, u_n, v_n) = \\ = \left\{ \begin{array}{ll} 1, & \text{если } u_n \leqslant a \text{ и } v_n \geqslant a \\ 0 & \text{в противном случае} \end{array} \right. : a \in (0, 1]\}. \tag{13}$$

$$\mathcal{F} = \langle F_5, G_8 \cup G_9 \cup G_{10} \cup G_{11} \cup G_{12}\rangle. \tag{14}$$

Справедлива следующая теорема [25, 30].

Теорема 7. *Пусть вероятностная мера* $\mathbf{P}$ *определяется ограниченной функцией плотности распределения с ограниченными частными производными первого порядка,* I *— ЗИП типа* S_{intn}*,* $\mathcal{F}$ *— базовое множество, задаваемое соотношениями (8)–(14). Тогда*

$$0 < T(I, \mathcal{F}, (4\,k + 2 + (1 + 6\,[\log_2 k]) \cdot c)\,(k\,(k+1)/2)^{n-1}) - R(I) \leqslant 4\,n + 1,$$

где c *— константа, зависящая от функции плотности распределения.*

Для равномерной вероятностной меры $c = 2$.

Этот результат был получен с использованием методов оптимальной декомпозиции и снижения размерности.

Приведем описание метода оптимальной декомпозиции применительно к одномерной задаче интервального поиска. Пусть нам дано множество $V = \{y_1, \ldots, y_k\}$, в котором мы должны производить поиск. Введем натуральное число m, являющееся параметром алгоритма. Если известна оценка сверху c функции плотности вероятности появления запросов (то есть $p(x) \leqslant c$), то в качестве параметра m возьмем $m = 2\,c\,[\log_2 k]$, если же c неизвестна, то вместо нее можно взять любое число, например, $c = 2$. Пусть $S = \{s_1, \ldots, s_m\}$, где $s_i = i/(m+1)$, $i = \overline{1,m}$. Производим предобработку, заключающуюся в сортировке множества V в порядке возрастания и построении множества $L = \{l_1, \ldots, l_m\}$, где l_i — целое число, являющееся номером максимальной записи из V, не большей, чем s_i, причем если такой записи не существует, то примем $l_i = 0$ ($i = \overline{1,m}$). Теперь поиск по произвольно взятому интервалу-запросу $x = (u, v)$ производится следующим образом.

Сначала вычисляется длина запроса x.

Если она меньше, чем $1/m$, то в множестве V бинарным поиском находится ближайшая справа к точке u запись. Далее, начиная с этой записи, просматриваются слева направо все записи из V и сравниваются с правым концом запроса — точкой v до тех пор, пока очередная запись не станет больше v. Тем самым в этом случае, помимо перечисления ответа, производится порядка $\log_2 k$ действий.

Если $v - u \geqslant 1/m$, то вычисляем номер $j =]u \cdot m[$ точки s_j, попадающей в интервал $[u, v]$. Теперь, начиная с записи с номером l_j, просматриваем справа налево записи из V и сравниваем с левым концом запроса — точкой u. Как только очередная запись окажется меньше u, мы, начиная с записи с номером $l_j + 1$, просматриваем слева направо записи из V и сравниваем с правым концом запроса — точкой v до тех пор, пока очередная запись не станет больше v. Тем самым в этом случае мы, помимо перечисления ответа, производим 4 лишних действия (сравниваем $v - u$ с $1/m$, вычисляем функцию $]u \cdot m[$, делаем 1 лишнее действие, идя справа налево, и 1 лишнее действие, идя слева направо).

Здесь множество L определяет точки разбиения на подзадачи, а каждая из подзадач является одномерной задачей о доминировании, которая, согласно теореме 4, решается очень просто.

Осталось заметить, что параметр m подобран так, что средняя сложность первого случая не превышает 1, если известна оценка сверху функции плотности вероятности, и не превышает некоторой константы, если эта оценка точно не известна. Поскольку вероятность множества запросов, длина которых не больше $1/m$, не превышает $2c/m$.

И, наконец, заметим, что данный алгоритм требует дополнительную память порядка $\log_2 k$, чтобы хранить множество L, в худшем случае время его поиска равно $\log_2 k$ плюс время перечисления ответа, а в среднем — совсем небольшая константа (приблизительно 5) плюс перечисление ответа.

4. Влияние на оптимальное решение главных параметров модели

Как можно видеть, все рассмотренные задачи в некотором смысле хорошие, а именно все допускают снижение среднего времени поиска фактически до минимума. Возникает вопрос: насколько устойчиво свойство «хорошести», названное каноническим эффектом, при вариации параметров задач поиска? К параметрам, которые можно варьировать в задачах поиска, можно отнести следующие:

- базовое множество функций, характеризующее набор доступных средств;
- ограничения на объем ИГ, характеризующий объем памяти, соответствующего ИГ алгоритма поиска;
- ε-расширение запроса; этот параметр позволяет получать вообще говоря новые типы задач поиска и применим к классу задач, которые можно условно назвать непрерывными (к нему относятся задача о доминировании, задача интервального поиска и задача поиска идентичных объектов, когда пространство запросов, например, — компактное подмножество вещественной

прямой) и состоит в том, что запрос в новой задаче получается ε-расширением запроса старой задачи.

Как и следовало ожидать, сложность задачи поиска существенно зависит от выбора базового множества. Причем часто можно получить весь спектр, начиная от перебора (как самого сложного) до алгоритмов, сложность которых практически совпадает с мощностной нижней оценкой. Проиллюстрируем этот тезис на примере одномерной задачи интервального поиска [32].

Теорема 8. *Если* $F_0 = \{\chi_a : a \in [0,1]\}$,

$$\chi_a(u,v) = \begin{cases} 1, & \text{если } u \leqslant a,\ v \geqslant a \\ 0, & \text{в противном случае} \end{cases},$$

$\mathcal{F}_0 = \langle F_0, \varnothing \rangle$, *то для произвольной ЗИП* $I = \langle X_{int1}, V, \rho_{int1} \rangle$, *такой, что все записи в библиотеке* V *различны, справедливо* $T(I, \mathcal{F}_0) = |V|$.

Этот результат означает, что если базовое множество состоит только из характеристических функций записей, то перебор является оптимальным алгоритмом.

Теорема 9. *Если* $\mathcal{F}_1 = \langle F_1 \bigcup F_2, \varnothing \rangle$, *и функция плотности распределения вероятностей* $p(u,v)$, *определяющая меру* $\mathbf{P}$ *вероятностного пространства над множеством запросов* X_{int1}, *ограничена, то для произвольной ЗИП* $I = \langle X_{int1}, V, \rho_{int1} \rangle$ *выполнено* $T(I, \mathcal{F}_1) - R(I) \leqslant \underline{O}(\sqrt{k})$ *при* $k \to \infty$, *где* $k = |V|$, *причем существуют такая вероятностная мера* $\mathbf{P}$ *и такая ЗИП* $I = \langle X_{int1}, V, \rho_{int1} \rangle$, *где* $|V| = k$, *что* $T(I, \mathcal{F}_1) - R(I) = \underline{O}(\sqrt{k})$ *при* $k \to \infty$.

Теорема 10. *Если* $\mathcal{F}_2 = \langle F_1 \bigcup F_2, G_3 \rangle$, *то для произвольной ЗИП* $I = \langle X_{int1}, V, \rho_{int1} \rangle$ *выполняется* $T(I, \mathcal{F}_2) - R(I) \leqslant] \log_2 k [$.

Теорема 11. *Если* $\mathcal{F}_3 = \langle F_1 \bigcup F_2, G_2 \bigcup G_3 \rangle$, *и функция плотности распределения вероятностей* $p(u,v)$, *определяющая меру* $\mathbf{P}$ *вероятностного пространства над множеством запросов* X_{int1}, *такая, что* $p(u,v) \leqslant c = \text{const}$, *то для произвольной ЗИП* $I = \langle X_{int1}, V, \rho_{int1} \rangle$, *такой, что* $|V| = k$, *выполняется* $T(I, \mathcal{F}_3) - R(I) \leqslant] \log \log_2 k [+ 6 + 2c$.

Теорема 12. *Если* $\mathcal{F}_4 = \langle F_1 \bigcup F_2, G_1 \bigcup G_2 \bigcup G_3 \rangle$ *и функция плотности распределения вероятностей ограничена, то для произвольной ЗИП* $I = \langle X_{int1}, V, \rho_{int1} \rangle$ *выполняется* $T(I, \mathcal{F}_4) - R(I) \leqslant 5$.

Зависимость сложности задачи поиска от объема памяти более «плавная», чем от базового множества. В качестве примера этой зависимости можно рассмотреть случай, когда задача поиска есть задача поиска идентичных объектов [25].

Теорема 13. *Пусть* $I = \langle X, V, \rho_{id} \rangle$ *— задача поиска идентичных объектов,* $|V| = k$, $\mathcal{F}$ *— базовое множество, задаваемое соотношениями* (1)–(3), c *— константа, ограничивающая функцию плотности распределения запросов,*

$$L_1(l) = \begin{cases} 0, & \text{если } l = 0 \\ ]\log_2 l[+1, & \text{если } l = 1, 2, 3 \\ \log_2 l + 2, & \text{если } l \geqslant 4 \end{cases} —$$

функция, определенная на множестве целых неотрицательных чисел. Тогда

$$1 < T(I, \mathcal{F}, 2 \cdot k + m - 1) \leqslant \\ \leqslant \frac{c}{m} \left(\left(k - \left[\frac{k}{m} \right] \cdot m \right) \cdot L_1 \left(\left[\frac{k}{m} \right] + 1 \right) + \right. \\ \left. + \left(m - k + \left[\frac{k}{m} \right] \cdot m \right) \cdot L_1 \left(\left[\frac{k}{m} \right] \right) \right) + 1.$$

В частности,

$$1 < T(I, \mathcal{F}, (2 + c) \cdot k) < 2$$

и $T(I, \mathcal{F}) \sim 1$ *при* $k \to \infty$.

Можно видеть, что при объеме памяти $2k$ мы имеем логарифмический поиск, а при увеличении объема до $(2+c)k$ мы плавно снижаем среднее время поиска до 2 операций. Эта зависимость более наглядна в асимптотической записи в случае равномерной вероятности запросов, то есть когда $c = 1$:

$$T(I, \mathcal{F}, 2k + m) \lesssim 2 + \log_2 k - \log_2 m.$$

Эта формула «разумна» при $0 \leqslant m \leqslant k$. Таким образом, в данной ситуации выигрыш по времени логарифмически зависит от приращения объема.

Если через k обозначить мощность библиотеки, то для двумерной задачи интервального поиска объем памяти, необходимый алгоритму, на котором достигается оценка теоремы 7, равен $\underline{O}(k^3)$. С целью понижения объема памяти в [33] 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 14. *Пусть I — двумерная задача интервального поиска, вероятностная мера $\mathbf{P}$ 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*

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

Тем самым здесь также наблюдается «плавная» зависимость сложности задачи поиска от объема памяти.

Ситуация, возникающая при обобщении задач за счет ε-расширения запроса, не однозначна. Так, в задачах о доминировании и интервального поиска при малых ε результаты, описанные в теоремах 6 и 7, полностью сохраняются, так как ε-расширение приводит лишь к вымыванию «малых» запросов, а поскольку их доля мала, то это не отражается на результате. В случае задачи поиска идентичных объектов в геометрической интерпретации, когда множество запросов есть отрезок $[0, 1]$ вещественной прямой, картина более интересная. При малых ε (например, при $\varepsilon < 1/k^2$, где k — мощность библиотеки) справедлива ситуация, описанная в теореме 13. А при больших ε задача превращается в упрощенную версию одномерной задачи интервального поиска, и результат будет аналогичен результату, описанному в теореме 12.

5. Заключение

На основе информационно-графовой модели можно предложить новую технологию проектирования физической организации баз данных (БД). По этой технологии на начальном этапе выделяются классы однотипных вопросов к БД, оформляемые в виде типов задач поиска. Множество данных БД задает конкретную задачу поиска данного типа. Для каждой задачи поиска выделяется множество элементарных операций над запросами, оформляемое в виде базового множества, и решается задача синтеза оптимального информационного графа, решающего данную задачу поиска. Полученный информационный граф описывает оптимальную структуру данных, соответствующую заданным целям оптимизации (среднему времени поиска, времени поиска в худшем случае, объему памяти).

Тем самым, один информационный граф описывает структурную часть БД, обрабатывающую один класс однотипных вопросов к БД. А сама БД в информационно-графовой модели представляется как совокупность нескольких информационных графов, охватывающих весь спектр вопросов к базе данных.

Поскольку в работе рассматриваются наиболее распространенные типы задач поиска, решение проблемы оптимального синтеза для данных задач позволяет для большинства случаев, возникающих при проектировании физической организации баз данных, иметь готовые рекомендации.

Среди основных направлений развития данной теории можно выделить следующие.

Во-первых, дальнейшее исследование как перечисленных выше типов задач поиска таких, например, как задачи поиска идентичных объектов (Луговская Ю. П. [26]), включающего (Косолапов А. В. [34]) и интервального (Ерохин А. Н. [35], Кузнецова И. В. [33, 36, 37]) поиска, так и новых типов задач поиска таких, например, как задача о метрической близости на булевом кубе (Быченкова Е. С. [38]) и в евклидовом пространстве (Гусманова Г. Ф. [39]), интервального поиска на булевом кубе (Блайвас Т. Д. [40]), задача о протыкании и многие другие. При этом хотелось бы выделить направление, связанное с исследованием функционнальной сложности информационных графов

(Кузнецова И. В. [33], Гусманова Г. Ф. [39]), то есть функции зависимости времени поиска от объема доступной памяти. Эти исследования очень полезны для практики, так как позволяют в зависимости от имеющихся ресурсов памяти подбирать наиболее быстрые алгоримы поиска.

Перспективными представляются направления, связанные с исследованием параллельных (Ерохина Е. Р. [28]), фоновых (Мхитарова Т. В. [31]) и нечетких (Фещук А. А. [41, 42]) задач поиска. В этих направлениях пока сделаны только первые шаги, а именно введены соответствующие обобщения информационно-графовой модели и получены первые результаты.

Среди задач поиска в базах данных кроме задач на перечисление ответа, которые описывались выше, есть еще задачи на поиск представителя, когда достаточно найти один объект, удовлетворяющий запросу. Такие задачи интересны как сами по себе, так и как вспомогательный аппарат в фоновых задачах поиска. Исследования в этом направлении находятся на начальной стадии.

Интересным представляется исследования, связанные с реализацией поисковых операторов в виде интегральных схем, при этом информационные графы не только подсказывают структуру схем, но и позволяют подбирать наиболее удобную элементную базу.

Список литературы

[1] Шеннон К. Работы по теории информации и кибернетике. М.: Изд-во иностранной литературы, 1963.

[2] Лупанов О. Б. О синтезе некоторых классов управляющих систем // Проблемы кибернетики. 1963. **10**. С. 63–97.

[3] Андреев А. Е. Метод бесповторной редукции синтеза самокорректирующихся схем // ДАН СССР. 1985. **283**. № 2. С. 265–269.

[4] Кудрявцев В. Б. Функциональные системы. М.: Изд-во МГУ, 1982.

[5] Ершов А. П. О программировании арифметических операторов // ДАН СССР. 1958. **118**. С. 427–430.

[6] Кнут Д. Искусство программирования для ЭВМ. Сортировка и поиск. **3**. М.: Мир, 1978.

[7] Ли Д., Препарата Ф. Вычислительная геометрия. Обзор // Кибернетический сб. 1987. **24**. С. 5–96.

[8] Мартин Дж. Организация баз данных в вычислительных системах. М.: Мир, 1980.

[9] Ньюмен У. М., Спруэлл Р. Ф. Основы интерактивной машинной графики. М.: Мир, 1976.

[10] Препарата Ф., Шеймос М. Вычислительная геометрия: Введение. М.: Мир, 1989.

[11] Решетников В. Н. Алгебраическая теория информационного поиска // Программирование. 1979. № 3. С. 68–74.

[12] Селтон Г. Автоматическая обработка, хранение и поиск информации. М.: Советское радио, 1973.

[13] Ben-Or M. Lower bounds for algebraic computation trees // Proc. 15th ACM Annu. Symp. Theory Comput. (Apr. 1983) P. 80–86.

[14] Bentley J. L., Friedman J. H. Data structures for range searching // Comput. Surveys. 1979. **11**. P. 397–409.

[15] Bentley J. L., Maurer H. A. Efficient worst-case data structures for range searching // Acta Inform. 1980. **13**. P. 155–168.

[16] Bentley J. L., Stanat D. F. Analysis of range range searching in quad trees // Inform. Processing Lett. 1975. **3**. P. 170–173.

[17] Bolour A. Optimal retrieval algorithms for small region queries // SIAM J. Comput. 1981. **10**. P. 721–741.

[18] Chazelle B. M. Filtering search: a new approach to query-answering // Proc. 24th IEEE Annu. Symp. Found. Comput. Sci. (Nov. 1983). P. 122–132.

[19] Fredman M. L., Baskett F., Shustek J. An algorithm for finding nearest neighbors // IEEE Trans. Comput. 1975. **C-24**. P. 1000–1006.

[20] Fredman M. L., Bentley J. L., Finkel R. A. An algorithm for finding best match in logarithmic expected time // ACM Trans. Math. Software. 1977. **3**. № 3. P. 209–226.

[21] 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. **9**. P. 23–29.

[22] Lueker G. S. A data structure for orthogonal range queries // Processing of the 19th Annual IEEE Symposium on Foundations of Computer Science. 1978. P. 28–34.

[23] Гасанов Э. Э. Об одной математической модели информационного поиска // Дискретная математика. 1991. **3**. № 2. С. 69–76.

[24] Гасанов Э. Э. Нижняя оценка сложности информационных сетей для одного класса задач информационного поиска // Дискретная математика. 1992. **4**. № 3. С. 118–127.

[25] Гасанов Э. Э. Мгновенно решаемые задачи поиска // Дискретная математика. 1996. **8**. № 3. С. 119–134.

[26] Гасанов Э. Э., Луговская Ю. П. Константный в худшем случае алгоритм поиска идентичных объектов // Дискретная математика. 1999. **11**. № 4. С. 139–144.

[27] Гасанов Э. Э. Оптимальные информационные сети для отношений поиска, являющихся отношениями линейного квазипорядка // Конструкции в алгебре и логике. Тверь: Изд-во Тверского гос. ун-та, 1990. С. 11–17.

[28] Гасанов Э. Э., Ерохина Е. Р. Моделирование и сложность поиска в многопроцессорных системах // Дискретная математика. 1999. **11**. № 3. С. 63–82.

[29] Гасанов Э. Э. Нижняя оценка сложности информационных сетей для одного отношения частичного порядка // Дискретная математика. 1996. **8**. № 4. С. 108–122.

[30] Гасанов Э. Э. Функционально-сетевые базы данных и сверхбыстрые алгоритмы поиска. М.: Изд. центр РГГУ, 1997.

[31] Гасанов Э. Э., Мхитарова Т. В. Об одной математической модели фоновых алгоритмов поиска и быстрый фоновый алгоритм двумерной задачи о доминировании // Фундаментальная и прикладная математика. 1997. **3**. № 3. С. 759–773.

[32] Гасанов Э. Э. Об одномерной задаче интервального поиска // Дискретная математика. 1995. **7**. № 2. С. 40–60.

[33] Гасанов Э. Э., Кузнецова И. В. О функциональной сложности двумерной задачи интервального поиска // Дискретная математика. В печати.

[34] Гасанов Э. Э., Косолапов А. В. К вопросу о древовидности оптимальных информационных сетей включающего поиска // Интеллектуальные системы. 1998. **3**. № 1–2. С. 167–192.

[35] Гасанов Э. Э., Ерохин А. Н. О быстром в среднем решении n-мерной задачи интервального поиска // Методы и системы технической диагностики (Тезисы X международной конференции по проблемам теоретической кибернетики). Саратов: Изд-во Саратовского университета, 1993. С. 48–49.

[36] Гасанов Э. Э., Кузнецова И. В. Оценки функциональной сложности двумерной задачи интервального поиска // Тезисы докладов XII Международной конференции «Проблемы теоретической кибернетики». Нижний Новгород, (17–22 мая 1999 г.) С. 47.

[37] Gasanov E. E., Kuznetsova I. V. On one method to decrease average search time // Abstracts of Ist Turkish World Mathematics Symposium. Elazig, Turkey. (29 June – 2 July 1999). P. 135.

[38] Быченкова Е. С. Асимптотическое решение задачи о метрической близости для одного базового множества функций. В печати.

[39] Гасанов Э. Э., Гусманова Г. Ф. Оценки функциональной сложности задачи о метрической близости в евклидовом пространстве. В печати.

[40] Блайвас Т. Д. Оптимальное решение задачи интервального поиска на булевом кубе в классе сбалансированных древовидных схем. В печати.

[41] Гасанов Э. Э., Фещук А. А. Информационно-графовая модель данных с нечеткой логикой // Труды IV Международной конференции по математическому моделированию, Москва (27 июня – 4 июля 2000 г.) Т. II. С. 16–20. М.: Изд-во «Станкин», 2001.

[42] Фещук А. А. К вопросу анализа нечетких информационных графов. В печати.