Информационно-графовая модель хранения и поиска данных *

Э.Э.Гасанов
Московский государственный университет

Аннотация

В теории баз данных исследуется множество разрозненных задач поиска. В работе предлагается новый подход, позволяющий с общих позиций взглянуть на эти задачи. Он основывается на теории управляющих систем и предлагает новую концепцию хранения данных и новый подход к поиску информации. В рамках этого подхода становится возможным исследование такой традиционно трудной меры сложности как среднее время поиска. В работе ставятся несколько базовых проблем поиска информации и приводится решение проблемы оптимального синтеза для этих базовых проблем. Исследуется также влияние на оптимальное решение таких факторов как объем памяти, базовое множество допустимых функций и ε-расширение запроса задачи.

1 Введение

В теории баз данных накоплено большое количество локальных примеров и задач поиска информации (в частности, в работе приводятся 7 основных модельных классов задач информационного поиска (ЗИП)), процесс этого накопления продолжается и в настоящее время. Эти примеры имеют разрозренный характер, а процесс накопления напоминает

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (грант 98-01-00130)

разрастание вширь, отражая каждый раз все новые особенности примеров. Вместе с тем естественно стремиться к тому, чтобы можно было усмотреть в них некое объединяющее начало, которое позволило бы оперировать накопленными примерами с единных позиций. Интенсивное параллельное развитие теории управляющих систем позволило разработать серьезный математический аппарат и накопить важные результаты по синтезу оптимальных схем. И хотя разработанный аппарат и накопленные здесь результаты не имеют прямого отношения к теории баз данных и информационного поиска, но могут служить хорошим примером и ассоциированным источником для развития данного направления.

В связи с имеющимся положением в теории баз данных, напрашивается определенная аналогия с ситуацией, которая возникла в теории синтеза управляющих систем. Там в роли модельных многообразий, подобных 7 модельным классам задач поиска, рассматриваемых в данной работе, выступают следующие: контактно-вентильные схемы, формулы, схемы из функциональных элементов, нормальные формы и др. [1, 2].

В теории синтеза основная задача долгое время состояла в следующем. Предполагалось, что на каждом модельном многообразии определена некоторая своя мера сложности, и задача оптимального синтеза состояла в том, чтобы для заданной булевской функции указать ту схему из заданного многообразия, которая реализует эту функцию и имеет минимальную или близкую к ней сложность. Позже [3] было установлено, что на самом деле эти сложностные характеристики могут быть извлечены из внутренних свойств самой функции.

В нашем случае в каждом из семи рассмотренных модельных случаев также изобретались свои сложностные характеристики и требовалось для ЗИП находить в определенном смысле простейшего представителя многообразия решений ЗИП. При этом в отличие от случая синтеза управляющих систем понятие сложности ЗИП не было единообразным.

Таким образом, в нашем случае для решения проблемы поиска информации актуальными становились сначала точная постановка этой проблемы, а затем и решение ее.

В постановочной части прежде всего необходимо было выработать общее понятие управляющей системы, адекватно характеризующей задачу поиска информации и процесс ее решения, в частности, охватывающее наши 7 случаев. Во-вторых, необходимо введение понятия сложности такой управляющей системы, отвечающей содержательным представлени-

ям.

После осуществления и того и другого возникают соответствующие проблемы анализа и синтеза указанных систем, оптимально реализующих заданный класс задач информационного поиска.

Мы предлагаем новую управляющую систему, названную информационным графом, которая в общей иерархии теории управляющих систем находится в не очень высоких слоях залегания и является в некотором смысле обощением 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 баз данных.

Информационный граф, который представляет собой ориентированный граф, ребра и вершины которого нагруженны функциями и элементами данных, с одной стороны дает новую концепцию хранения данных, а с другой стороны предлагает новый подход к поиску информации, как волнового процесса на графах, управляемого нагрузочными функциями. Нагрузочные функции, которые называются базовыми, разделены на два класса — предикаты и переключатели (первые приписываются ребрам графа, а вторые — вершинам), и являются одним из основных управляющих параметров модели.

Кроме того, 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 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.

Анализ рассматриваемых в работе 7 модельных классов задач поиска позволил 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. 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. 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.

Классификация 7 модельных классов задач поиска и объединение их в 3 базовых класса не единственный способ обобщения задач поиска. Одним из естественных методов является ε-расширение запроса, которое позволяет ответу на задачу несколько отходить (на величину ε) от требований задачи.

Для базовых задач поиска ставится 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. Результаты, полученные при решении проблемы оптимального синтеза для базовых задач, характеризуются, во-первых, тем, что для всех задач из модельных классов, кроме так называемых задач включающего поиска из второго базового класса, получены точные и (или) асимптотические результаты, а для задач включающего поиска получена асимптотика функции Шеннона и асимпотика логарифма сложности для почти всех задач и для средней сложности по задачам. Во-вторых, полученные результаты можно условно разбить на 4 типа.

К первому типу относятся задачи, имеющие константную сложность, т.е. которые можно решить в среднем за константное число шагов. Константную сложность имеют некоторые задачи из первого базового класса.

Ко второму типу относятся задачи поиска с условно константной сложностью. Это задачи, для решения которых помимо перечисления ответа (а это сложность, которую избежать никак нельзя, и она носит название мощностной нижней оценки) требуется в среднем константное

число вычислений. Такие задачи встречаются во втором и третьем базовых классах.

К третьему типу относятся задачи с логарифмической сложностью, решение которых не может быть получено за время, меньшее, чем логарифм от объема базы данных. К этому типу относятся некоторые задачи из первого базового класса, когда из множества базовых функций исключены переключатели.

И, наконец, к четвертому типу относятся задачи с быстро растущей сложностью, т.е. разность между сложностью которых и мощностной нижней оценкой является растущей функцией. К четвертому типу относятся задачи включающего поиска из второго базового класса.

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 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.

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 (т.е. число ребер информационного графа), множество функций, которые разрешается использовать при решении (т.е. множество базовых функций, используемых при нагрузке графа), и ε-расширение запроса. Показывается, что при любой вариации, кроме ε-расширения запроса при доста-

точно малых ε, мы уходим от канонического эффекта.

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 Постановка базовых проблем

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, т.е. 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.

Для исследования были выбраны 7 основных классов ЗИП, играющих роль модельных объектов.

1) ЗИП, в которых вероятность множества запросов, ответ на которые содержит более c записей ($c = const$), равна нулю;

2) поиск идентичных объектов, когда в библиотеке надо найти записи равные запросу;

3) задачи о близости, которые состоят в поиске в линейно-упорядоченном множестве объекта, ближайшего к объекту-запросу;

4) ЗИП, когда отношение поиска является отношением частичного порядка, заданное на конечном множестве, в частности, включающий поиск (это задача поиска на булевом кубе точек не больших по-компонентно чем запрос);

5) ЗИП, когда отношение поиска является отношением линейного предпорядка;

6) 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.

7) задача интервального поиска, которая состоит в поиске в конечном подмножестве n-мерного пространства всех тех точек, которые попадают в n-мерный параллелепипед-запрос;

Выбор модельных классов определяется как повсеместностью использования их в базах данных, так и частотой цитирования в литературе [4–21].

Нетрудно видеть, что все эти случаи 1–7 возникают как соответствующие многообразия ЗИП либо за счет фиксации свойств решения ЗИП, либо за счет дополнительных предположений относительно X, Y, V, ρ.

Анализ случаев 1–7 позволяет заметить, что все модельные задачи можно разбить на 3 класса. К первому классу, получившему название задач поиска с коротким ответом, относятся первые три модельные задачи. Во второй класс, названный задачами поиска на частично-упорядоченных множествах данных, попали четвертая, пятая и шестая модельные задачи. И седьмая задача образовала третий класс.

Итак, ставятся две главные задачи:

- разработка общего модельного объекта для исследования сложностных характеристик алгоритмов поиска информации, в частности алгоритмов решения модельных ЗИП 1–7;

- исследование в рамках данного модельного объекта для модельных классов ЗИП 1–7 такой сложностной характеристики, как среднее

время поиска, и изучение влияния на эту характеристику, таких управляющих параметров как объем памяти, имеющийся в распоряжении (т.е. число ребер информационного графа), множество функций, которые разрешается использовать при решении (т.е. множество базовых функций, используемых при нагрузке графа), и ε-расширение запроса.

3 Информационный граф — новая концепция хранения данных и новый подход к поиску информации

Опишем основной объект, который называется информационным графом (ИГ). Вводить ИГ мы будем, одновременно иллюстрируя его на примере одномерной задачи интервального поиска. 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:

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

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

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

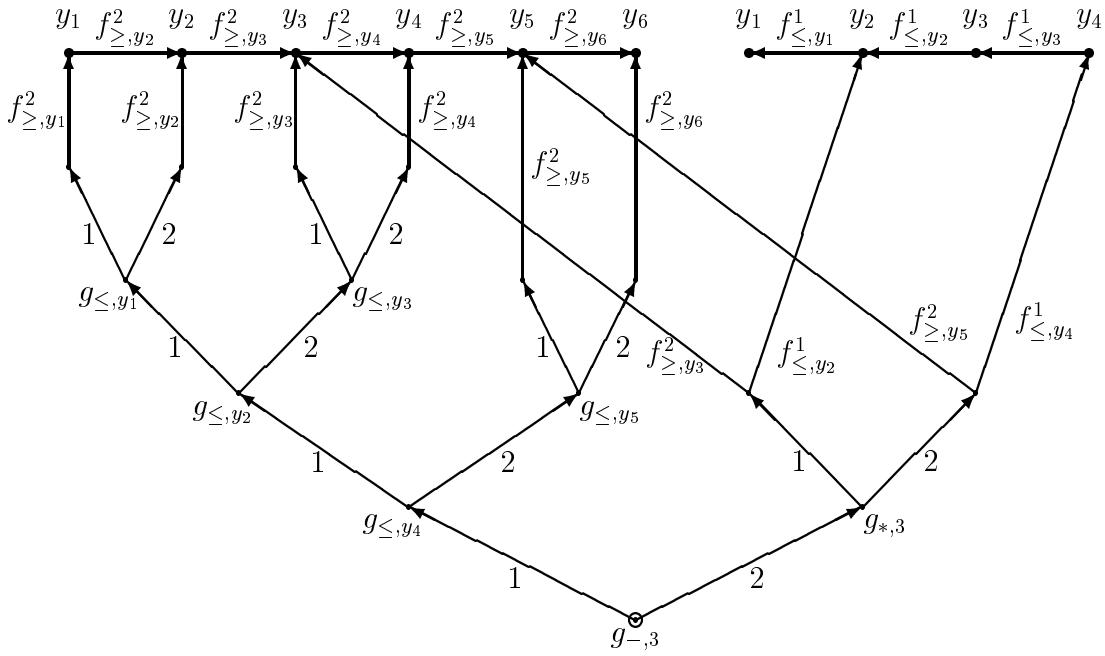

Рис. 1: Решение одномерной задачи интервального поиска

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

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

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

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 7). Ребра, исходящие из каждой из переключательных вершин, нумеруются начиная с 1 и называются

переключательными ребрами (на рисунке таких ребер 14). Ребра, не являющиеся переключательными, называются предикатными и им приписываются предикаты из множества F (на рисунке таких ребер 18). 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;

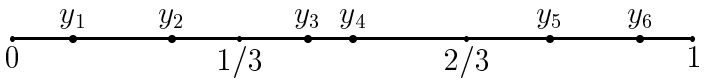

Рис. 2:

- 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.

Достоинства ИГ:

1) все основные известные ранее объекты, используемые для моделирования алгоритмов поиска, являются частными случаями ИГ;

2) все наиболее популярные и известные алгоритмы поиска легко перекладываются на язык ИГ и при этом приобретают свойство метризуемости, т.е. становится легко подсчитать такие характеристики сложности алгоритмов как объем памяти, время поиска в среднем и время поиска в худшем случае, причем полученные с помощью ИГ оценки полностью согласуются с оценками полученными другими методами, если такие оценки существовали.

Подтверждение сказанного:

1) Наиболее популярная модель, используемая для оценки сложности алгоритмов, — алгебраическое дерево вычислений (АДВ-модель), а также ее разновидность — алгебраическое дерево решений, при переводе на язык ИГ описываются некоторым классом древовидных ИГ и тем самым являются частным случаем ИГ.

2) Алгоритмы и конструкции, используемые в древовидных и лингвистических базах данных, описываются древовидными ИГ.

3) Алгоритм поиска, используемый в дедуктивных базах данных, при переходе на язык ИГ приводит к константному дереву.

4) Алгоритмы в реляционных базах данных приводят к деревовидным ИГ специального вида.

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

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

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.

Возникает стандартная проблема синтеза: по заданной ЗИП построить ИГ над некоторым базовым множеством, разрешающий данную ЗИП, и соответственно вопрос о полноте базового множества.

Скажем, что базовое множество $\mathcal{F}$ *полно* для типа $S = \langle X, Y, \rho \rangle$, если для любой ЗИП $I = \langle X, V, \rho \rangle$ типа S существует ИГ U над базовым множеством $\mathcal{F}$, разрешающий ЗИП I.

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 $y \in Y$.

Характеристической функцией записи y назовем функцию

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

Если n — натуральное число, а $g(x)$ — некий переключатель, то обозначим

$$\xi_g^n(x) = \begin{cases} 1, & \text{если } g(x) = n \\ 0, & \text{если } g(x) \neq n \end{cases}.$$

Обозначим

$$\widehat{G} = \{\xi_g^n : g \in G, n \in \mathbf{N}\},$$

где $\mathbf{N}$ — множество натуральных чисел.

Справедлив следующий результат, относящийся к проблеме полноты для ИГ [22].

Теорема 1 *Пусть заданы множества запросов X, записей Y и отношение поиска ρ на $X \times Y$. Тогда базовое множество $\mathcal{F} = \langle F, G\rangle$ будет полным для типа $S = \langle X, Y, \rho\rangle$ тогда и только тогда, когда для любой записи $y \in Y$ такой, что $O(y, \rho) \neq \emptyset$, функцию $\chi_{y,\rho}(x)$ можно представить формулой вида*

$$\chi_{y,\rho}(x) = \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} f_{ij}(x),$$

где $f_{ij} \in F \cup \widehat{G}$.

4 Сложность поиска в информационных графах

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 $\mathcal{F} = \langle F, G\rangle$ измеримое, если алгебра σ содержит все множества N_f, где $f \in F \cup \widehat{G}$.

Справедливо утверждение, что если базовое множество $\mathcal{F}$ измеримое, то для любого ИГ U над базовым множеством $\mathcal{F}$ функция $T(U,x)$ как функция от x является случайной величиной.

Далее всюду будем предполагать, что базовое множество измеримое.

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

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

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* $\mathcal{F}$.

Справедлива следующая теорема [22].

Теорема 2 (мощностная нижняя оценка) *Если* $I = \langle X, V, \rho\rangle$ *— произвольная ЗИП,* $\mathcal{F}$ *— измеримое базовое множество, такое, что множество* $\mathcal{U}(I,\mathcal{F}) \neq \emptyset$, *то*

$$T(I,\mathcal{F}) \ge \sum_{y\in V} \mathbf{P}(O(y,\rho)).$$

Этот результат был получен с помощью метода характеристических носителей графа.

5 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** ($c = const$) **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 [23].

Теорема 3 *Если* $I = \langle X, V, \rho \rangle$ *— ЗИП, обладающая* G*-свойством,* $\mathcal{F} = \langle F, \emptyset \rangle$ *— некоторое специальное базовое множество, то*

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

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 информационном массиве объекта, идентичного объекту-запросу. Формально она принадлежит следующему типу: $S_{id} = \langle X, X, \rho_{id} \rangle$, где отношение поиска ρ_{id} есть отношение идентичности, то есть $x \rho_{id} y \iff x = y$. Справедлива теорема [24].

Теорема 4 *Пусть* $I = \langle X, V, \rho_{id} \rangle$ *— задача поиска идентичных объектов, то есть задача типа* S_{id}*, где* $|V| = k$*,* $\mathcal{F}$ *— некоторое измеримое базовое множество, c — константа, ограничивающая функцию плотности распределения запросов. Тогда* $1 < T(I, \mathcal{F}, (2+c)\cdot k) < 2$ *и* $T(I, \mathcal{F}) \sim 1$ *при* $k \to \infty$*.*

Здесь под словом "некоторое"предполагается, что базовое множество содержит функции сравнения и разбиения множества X на приблизительно равные части (умножение).

Приводится также алгоритм, который в типичной ситуации при затратах памяти k^2 обеспечивает время поиска, в худшем случае равное 2 [25].

Задачи о близости состоят в поиске в линейно-упорядоченном множестве объекта, ближайшего к объекту-запросу. В задачах о близости (ЗоБ) в отличие от ЗИП отношение поиска задается не на $X \times Y$, а на $X \times V$, где V — библиотека задачи о близости.

Рассмотрим следующую задачу о близости. Пусть на множестве записей Y задано отношение линейного порядка $\preceq$.

Отношение поиска ρ_{near1} задается на $X \times V$ и определяется соотношением

$$x\rho_{near1}y \Longleftrightarrow (y \in V)\&(x \preceq y)\&(\neg(\exists y')((y' \in V)\&(x \preceq y')\&(y' \prec y))),$$

т.е. $x\rho_{near1}y$, если $y \in V$, ближайшее справа к x.

При выполнении этих условий ЗИП $I = \langle X, V, \rho_{near1} \rangle$ назовем первой задачей о близости. Справедлива следующая теорема [24].

Теорема 5 *Пусть* $I = \langle X, V, \rho_{near1} \rangle$ *— первая задача о близости, где* $|V| = k$*,* $\mathcal{F}$ *— некоторое измеримое базовое множество, c — константа, ограничивающая функцию плотности распределения запросов. Тогда* $1 < T(I, \mathcal{F}, (2 + c) \cdot k + 1) < 2$ *и* $T(I, \mathcal{F}) \sim 1$ *при* $k \to \infty$*.*

Последние две теоремы получены методом оптимальной декомпозиции.

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 [26].

Теорема 6 *Для любой ЗИП $I = \langle X, V, \overset{l}{\succeq} \rangle$ типа S_{lin} существует оптимальный ИГ над базовым множеством $\mathcal{F} = \langle \mathcal{K}, \emptyset \rangle$ и*

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

Для ЗИП типа S_{lin} исследовалось также параллельное решение [27], которое предполагает, что ИГ обрабатывается сразу несколькими вычислителями, при этом выделяется 2 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

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

причем на B^n задана равномерная вероятностная мера, т.е. для $\forall x \in B^n$ $\mathbf{P}(x) = 1/2^n$ и $\forall A \subseteq B^n$ $\mathbf{P}(A) = |A|/2^n$. Справедлива следующая теорема [28].

Теорема 7 *Пусть базовое множество имеет вид $\mathcal{F} = \langle F, \emptyset \rangle$, 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}

$$T(I, \mathcal{F}) \geq 2 \cdot \sum_{y \in V} \mathbf{P}(O(y, \overset{b}{\succeq}))$$

и существуют такие ЗИП $I = \langle B^n, V, \overset{b}{\succeq} \rangle$ *типа* S_{bool}, *что*

$$T(I, \mathcal{F}) = 2 \cdot \sum_{y \in V} \mathbf{P}(O(y, \overset{b}{\succeq}))(1 + \bar{o}(1))$$

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, т.е. 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 7 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.

Опишем тип задач поиска, который соответствует n-мерной задаче о доминировании.

Пусть $Y_{dom} = [0,1]^n$ — множество записей и $X_{dom} = [0,1]^n$ — множество запросов. Пусть на множестве X_{dom} задано вероятностное пространство $\langle X_{dom}, \sigma, \mathbf{P} \rangle$, где $\mathbf{P}$ задается функцией плотности вероятности $p(x)$.

Отношение поиска ρ_{dom} определено на $X_{dom} \times Y_{dom}$ и задается следующим соотношением:

$$(x_1, x_2, \ldots, x_n)\rho_{dom}(y_1, y_2, \ldots, y_n) \Longleftrightarrow y_i \leq x_i,\ i = \overline{1, n}.$$

Тогда тип

$$S_{dom} = \langle X_{dom}, Y_{dom}, \rho_{dom}, \sigma, \mathbf{P} \rangle$$

назовем типом задачи о доминировании. Справедлива следующая теорема [29].

Теорема 8 *Пусть ЗИП* $I = \langle X_{dom}, V, \rho_{dom} \rangle$ *— n-мерная задача о доминировании, то есть задача типа* S_{dom}*, где* $|V| = k$*. Пусть* $\mathcal{F}$ *— базовое множество, содержащее арифметические операции и операции сравнения,* $R(I) = \sum\limits_{y \in V} \mathbf{P}(O(y, \rho_{dom}))$*. Тогда, если функция плотности вероятности* $p(x) \leq c$*, то*

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

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 5) 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 (т.е. в среднем за $2(n - 1)$ вычислений) мы придем к одномерной задаче о доминировании, оптимальное решение которой приводится в теореме 6.

Для двумерной задачи о доминировании исследовалось также решение задачи в фоновом режиме [30]. 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 образует последний класс с тем же названием и состоит в поиске в конечном подмножестве n-мерного пространства всех тех точек, которые попадают в n-мерный параллелепипед-запрос.

Пусть $Y_{intn} = [0,1]^n$ — множество записей и

$$X_{intn} = \{\tilde{x} = (u_1, v_1, \ldots, u_n, v_n) : 0 \le u_i \le v_i \le 1,\ i = \overline{1,n}\} —$$

множество запросов. Пусть на множестве X_{intn} задано вероятностное пространство $\langle X_{intn}, \sigma, \mathbf{P}\rangle$, где $\mathbf{P}$ задается функцией плотности вероятности $p(\tilde{x})$.

Отношение поиска ρ_{intn} определено на $X_{intn} \times Y_{intn}$ и задается следующим соотношением:

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

Тогда тип

$$S_{intn} = \langle X_{intn}, Y_{intn}, \rho_{intn}, \sigma, \mathbf{P}\rangle$$

назовем типом интервального поиска. Справедлива следующая теорема [24, 29].

Теорема 9 *Пусть ЗИП* $I = \langle X_{intn}, V, \rho_{intn}\rangle$ *— n-мерная задача интервального поиска, то есть задача типа* S_{intn}*, где* $|V| = k$*. Пусть* $\mathcal{F}$ *— базовое множество, содержащее арифметические операции и операции сравнения,* c *— некоторая константа, определяемая по функции плотности вероятности* $p(\tilde{x})$*,* $R(I) = \sum\limits_{\tilde{y}\in V} \mathbf{P}(O(\tilde{y}, \rho_{intn}))$*. Тогда*

$$\begin{aligned} R(I) &< T(I, \mathcal{F}, (4\,k + 2 + (1 + 6\,[\log_2 k]) \cdot c)\,(k\,(k+1)/2)^{n-1}) \le \\ &\le R(I) + 4\,n + 1. \end{aligned}$$

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 $p(x) \leq c$), 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 $v - u \geq 1/m$, то вычисляем номер $j = \max(1,]u \cdot m[)$ 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 $\max(1,]u \cdot m[)$, 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 6 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.

6 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 [31].

Теорема 10 *Если* $F_0 = \{\chi_a : a \in [0,1]\}$,

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

$\mathcal{F}_0 = \langle F_0, \emptyset \rangle$, *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 11 *Если* $\mathcal{F}_1 = \langle F_1 \bigcup F_2, \emptyset \rangle$ *и функция плотности распределения вероятностей* $p(u,v)$, *определяющая меру* $\mathbf{P}$ *вероятностного пространства над множеством запросов* X_{int1}, *ограничена, то для произвольной ЗИП* $I = \langle X_{int1}, V, \rho_{int1} \rangle$

$$T(I, \mathcal{F}_1) - \sum_{y \in V} \mathbf{P}(O(y, \rho_{int1})) \le \underline{O}(\sqrt{k})$$

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*

$$T(I, \mathcal{F}_1) - \sum_{y \in V} \mathbf{P}(O(y, \rho_{int1})) = \underline{O}(\sqrt{k})$$

при $k \to \infty$.

Теорема 12 *Если* $\mathcal{F}_2 = \langle F_1 \bigcup F_2, G_3 \rangle$, *то для произвольной ЗИП* $I = \langle X_{int1}, V, \rho_{int1} \rangle$

$$T(I, \mathcal{F}_2) - \sum_{y \in V} \mathbf{P}(O(y, \rho_{int1})) \le]\log_2 k[.$$

Теорема 13 *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* $p(u,v) \leq c = const.$ *то для произвольной ЗИП* $I = \langle X_{int1}, V, \rho_{int1} \rangle$*, такой, что* $|V| = k$

$$T(I, \mathcal{F}_3) - \sum_{y \in V} \mathbf{P}(O(y, \rho_{int1})) \leq] \log \log_2 k [+ 6 + 2c.$$

Теорема 14 *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* $I = \langle X_{int1}, V, \rho_{int1} \rangle$

$$T(I, \mathcal{F}_4) - \sum_{y \in V} \mathbf{P}(O(y, \rho_{int1})) \leq 5.$$

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 [24].

Теорема 15 *Пусть* $I = \langle X, V, \rho_{id} \rangle$ *— задача поиска идентичных объектов,* $|V| = k$*,* $\mathcal{F}$ *— некоторое измеримое базовое множество, c — константа, ограничивающая функцию плотности распределения запросов,*

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

функция, определенная на множестве целых неотрицательных чисел. Тогда

$$\begin{aligned} 1 \quad &< \quad T(I, \mathcal{F}, 2 \cdot k + m - 1) \leq \\ &\leq \quad \frac{c}{m} \left(\left(k - \left[\frac{k}{m} \right] \cdot m \right) \cdot L_1 \left(\left[\frac{k}{m} \right] + 1 \right) + \right. \\ &\quad \left. + \left(m - k + \left[\frac{k}{m} \right] \cdot m \right) \cdot L_1 \left(\left[\frac{k}{m} \right] \right) \right) + 1. \end{aligned}$$

В частности,

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

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

Здесь как и в теореме 4 под словом "некоторое"предполагается, что базовое множество содержит функции сравнения и разбиения множества X на приблизительно равные части (умножение).

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, т.е. когда $c = 1$:

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

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

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 8 и 9, 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 15. А при больших ε задача превращается в упрощенную версию одномерной задачи интервального поиска и результат будет аналогичен результату, описанному в теореме 14.

Список литературы

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

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

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

[4] Ершов А. П. О программировании арифметических операторов. *ДАН СССР* (1958) **118**, 427–430.

[5] Кнут Д. *Искусство программирования для ЭВМ. Сортировка и поиск.* **3**, Мир, Москва, 1978.

[6] Ли Д., Препарата Ф. Вычислительная геометрия. Обзор. *Кибернетический сб.* (1987) **24**, 5–96.

[7] Мартин Дж. *Организация баз данных в вычислительных системах.* Мир, Москва, 1980.

[8] Ньюмен У. М., Спруэлл Р. Ф. *Основы интерактивной машинной графики.* Мир, Москва, 1976.

[9] Препарата Ф., Шеймос М. *Вычислительная геометрия: Введение.* Мир, Москва, 1989.

[10] Решетников В. Н. Алгебраическая теория информационного поиска. *Программирование* (1979), № 3, 68–74.

[11] Селтон Г. *Автоматическая обработка, хранение и поиск информации.* Советское радио, Москва, 1973.

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

[13] Bentley J. L., Friedman J. H. Data structures for range searching. *Comput. Surveys* (1979), **11** 397–409.

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

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

[16] Bolour A. Optimal retrieval algorithms for small region queries. *SIAM J. Comput.* (1981) **10**, 721–741.

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

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

[19] 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, 209–226.

[20] 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** 23–29.

[21] Lueker G. S. A data structure for ortogonal range queries. *Processing of the 19th Annual IEEE Symposium on Foundations of Computer Science.* (1978), 28–34.

[22] Гасанов Э. Э. Об одной математической модели информационного поиска. *Дискретная математика* (1991) **3**, № 2, 69–76.

[23] Гасанов Э. Э. Нижняя оценка сложности информационных сетей для одного класса задач информационного поиска. *Дискретная математика* (1992) **4**, № 3, 118–127.

[24] Гасанов Э. Э. Мгновенно решаемые задачи поиска. *Дискретная математика* (1996) **8**, № 3, 119–134.

[25] Гасанов Э. Э., Луговская Ю. П. Константный в худшем случае алгоритм поиска идентичных объектов. *Дискретная математика* (в печати).

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

[27] Гасанов Э. Э., Ерохина Е. Р. Моделирование и сложность поиска в многопроцессорных системах. *Дискретная математика* (в печати).

[28] Гасанов Э. Э. Нижняя оценка сложности информационных сетей для одного отношения частичного порядка. *Дискретная математика* (1996) **8**, № 4, 108–122.

[29] Гасанов Э. Э. *Функционально-сетевые базы данных и сверхбыстрые алгоритмы поиска.* Изд. центр РГГУ, Москва, 1997.

[30] 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, 759–773.

[31] Гасанов Э. Э. Об одномерной задаче интервального поиска. *Дискретная математика* (1995) **7**, № 2, 40–60.