

МИНИСТЕРСТВО ОБЩЕГО И ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Российский государственный гуманитарный университет

**Факультет защиты
информации**

Кафедра математической и
программной защиты информации

**Центр компьютерных
технологий образования**

Э. Э. Гасанов

Функционально-сетевые базы данных и сверхбыстрые
алгоритмы поиска

Конспект лекций

Москва 1997

УДК 517.977 + 681.3 513

Факультет защиты информации
Кафедра математической и программной защиты информации
Центр компьютерных технологий образования

Гасанов Э. Э.

Функционально-сетевые базы данных и сверхбыстрые алгоритмы поиска. Конспект лекций. – М.: Издательский центр РГГУ. 1997. 88 с.

Курс лекций "Оптимальный поиск в базах данных" читается на факультете защиты информации Российского государственного гуманитарного университета в качестве спецкурса для специальности №220600 "Организация и технология защиты информации". В книге описывается подход к исследованию сложности алгоритмов поиска, основанный на построении математической модели алгоритмов поиска. Приводятся сверхбыстрые в "среднем" алгоритмы поиска, используемые в геометрических базах данных.

Для математиков-прикладников, специалистов в области теории баз данных, защиты информации и т.д., а также для аспирантов и студентов вузов как учебное пособие по математической теории баз данных и теории быстрых алгоритмов поиска.

Автор: к.ф.-м.н., доц. Э. Э. Гасанов

Рецензент: проф. А. С. Строгалов

© Российский государственный
гуманитарный университет, 1997

Оглавление

Введение	4
1 Информационные сети как математическая модель алгоритмов поиска информации	9
1.1 Понятие информационной сети	9
1.2 Необходимое и достаточное условие допустимости информационных сетей .	12
1.3 К вопросу о полноте для информационных сетей	14
1.4 Понятие сложности информационных сетей	15
1.5 Тривиальная нижняя оценка	18
2 Мгновенно решаемые задачи поиска	21
2.1 Задачи поиска с отношением поиска, являющимся отношением линейного квазипорядка	22
2.2 Поиск идентичных объектов	23
2.3 Задачи о близости	29
2.4 Многомерная задача интервального поиска	31
2.4.1 Одномерный случай	35
2.4.2 Многомерный случай	39
2.4.3 Пример оценки константы специальной ограниченности	45
2.5 Задача о доминировании	47
2.5.1 Основные понятия и формулировка результата	47
2.5.2 Мгновенное решение задачи о доминировании	48
Литература	57

Считается почти нормальным, что теория всегда будет идти позади. Но все-таки, действительно, надо бы обращать внимание на более простые вопросы, не завихряясь вокруг сложностей, так чтобы ну хоть где-нибудь забежать вперед и протянуть практике руку помощи.

M.P. Шура-Бура¹

Введение

Управляющая система, функционирующая в среде, — это центральное понятие кибернетики.

Управляющие системы можно разбить на два класса: те, которые действуют без памяти (рефлекторно), и те, которые имеют память. Второй класс неизмеримо богаче, чем первый, и изучается в первую очередь. Примеры управляющих систем без памяти: вирусы, микробы, разменные аппараты. Высшие животные, человек, в технике — адаптивные системы, компьютеры — управляющие системы с памятью. Естественно возникает вопрос оптимальной организации памяти в таких системах, которую можно отождествить с содержательным пониманием баз данных.

Как следует из известных справочников [26, 30], *база данных* — это именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.

Если следовать Дж. Мартину [25], К. Дейту [19], Т. Тиори и Дж. Фраю [32], то организацию базы данных можно условно разделить на логическую и физическую. Логическая организация базы данных отражает представление о базе данных прикладного программиста и пользователя, а физическая — представление системного программиста и аналитика. В [32] выделяется еще один уровень, более абстрактный, чем логический, называемый концептуальным представлением и отражающий представление администратора базы данных. Общепринято, что основными моделями логической организации базы данных являются следующие три модели: иерархическая, сетевая и реляционная.

Самой известной системой, использующей иерархическую модель данных, является система IMS фирмы IBM [46, 54, 55, 56]. В иерархической модели отношения между данными бывают типа "родитель — потомки", то есть у каждого объекта только один родитель (у корневого объекта нет родителя), но в принципе может быть несколько потомков. Такие отношения принято изображать в виде дерева, где ребро между объектами отображает наличие некоторого отношения, причем название отношения пишется на ребре. Например, между объектами "клиент" и "заказ" может быть отношение, которое называется "делает", а между "заказ" и "товары" — отношение "состоит из".

В случае, когда граф отношений между объектами может представляться не только древовидными структурами, мы имеем дело с сетевой моделью данных, предложенной ассоциацией CODASYL [21, 36, 37, 38]. Понятно, что сетевая модель как более общая предоставляет большие возможности по сравнению с иерархической, но, с другой стороны, она сложнее в реализации и использовании.

Заслуга разработки и развития реляционной модели баз данных принадлежит Е. Кодду [39, 40, 41, 42, 43, 44, 45, 47]. Реляционная база данных состоит из плоских таблиц, называемых *отношениями*. Строки таблицы (экземпляры записей) называются *кортежами*, а столбцы — *доменами*.

¹Из дискуссии после доклада А.П.Ершова [20].

Для описания отношений и операций над ними существуют точные математические обозначения, основанные на алгебре отношений или на исчислении отношений. Е. Кодд [42] разработал специальный язык манипулирования данными для такой базы.

Различные пользователи могут выделять в базе данных различные наборы элементов данных и связи между ними. Следовательно, необходимо иметь возможность извлекать подмножества столбцов таблицы для одних пользователей, создавая таблицы меньшей размерности, а также объединять таблицы для других пользователей, создавая при этом таблицы большей размерности. Язык Е. Кодда содержит обе операции. Благодаря операциям разрезания и склеивания таблицы обладают гибкостью, которой лишено большинство древовидных и сетевых структур.

С логической точки зрения база данных — это множество двумерных таблиц с операциями извлечения и объединения столбцов.

С помощью метода нормализации Е. Кодда [40] можно древовидное или сетевое описание преобразовать к набору плоских таблиц.

При физической организации баз данных мы имеем дело не с представлением данных в прикладных программах, а с их размещением на запоминающих устройствах.

Критерии, определяющие выбор физической организации, отличаются от тех, которые определяют выбор логической организации данных. При выборе физической организации решающим фактором является эффективность, причем согласно Дж. Мартину [25] на первом месте стоит обеспечение эффективности поиска, далее идут эффективность операций занесения и удаления и затем обеспечение компактности данных. Кроме того, в последнее время большую актуальность приобрели проблемы защиты данных от несанкционированного доступа.

Методы физической организации данных хорошо изложены в классических книгах Д. Кнута [23], Дж. Мартина [25], А. Ахо, Дж. Хопкрофта, Дж. Ульмана [3].

Автор данной работы относит себя к разработчикам структур и алгоритмов физической организации данных. Так, предлагаемая в данной работе модель, предназначается для исследования структур и алгоритмов поиска информации. В этой модели, представляющей собой нагруженную ориентированную сеть, граф сети отражает структуру данных, а нагрузка сети функциями над множеством запросов предоставляет способы ориентации в данных, запирая или открывая проводимость ребер на конкретных запросах. Академик АТН РФ, профессор В.Б.Кудрявцев предложил более широко взглянуть на эту модель, подчеркнув, что ее можно рассматривать и как модель логической организации данных. В этой модели, которую академик В.Б.Кудрявцев предложил назвать функционально-сетевыми базами данных, граф сети, так же как и в сетевой модели отображает структуру и отношения между данными, а нагрузка ребер сети характеризует отношение между объектами, но в отличие от сетевой модели это отношение становится функциональным, то есть его содержание зависит от текущего запроса (или в более общем понимании от текущего состояния базы данных).

Представляется, что это настолько интересная и перспективная концепция, что ее название вынесено в заголовок книги.

Данная работа посвящена исследованию сложности алгоритмов, предназначенных для решения задач поиска. В понятие задачи поиска исследователи вкладывают по крайней мере 3 различных смысла.

Специалисты в теории исследования операций понимают задачи поиска как задачи управления сближением одной системы (поисковой) с другой (искомым объектом) по неполной априорной информации. Понимается, что цель поиска — это обнаружение искомого объекта, определяемое как выполнение определенных терминалльных условий. Ин-

тенсивно проблемой поиска подвижных объектов начали заниматься в период Второй мировой войны. Интерес к этой проблеме в тот период был вызван необходимостью разработки тактики борьбы против подводных лодок. Среди работ, посвященных этой тематике, можно выделить, например, работы В. А. Абчука, В. Г. Суздаля [1], О. Хеллмана [33] и Д. П. Кима [22].

Другое понимание задач поиска можно найти в книге Р.Альсведе, И.Вегенера "Задачи поиска"[2]. Приведем поясняющий пример из этой книги.

Во время Второй мировой войны все призывающие в армию США подвергались проверке на реакцию Вассермана. При этом проверялось, есть ли в крови обследуемого определенные антитела, имеющиеся только у больных. Во время этого массового обследования было замечено, что разумнее анализировать пробу крови целой группы людей. Если такая объединенная пробы крови не содержит антител, то, значит, ни один из обследуемых не болен. В противном случае среди них есть хотя бы один больной. Хороший алгоритм поиска для этой задачи — это тот, который для типичной выборки людей позволяет "наискорейшим образом" выявлять множество всех больных.

Другой пример мы находим в статье Д.Ли, Ф.Препараты "Вычислительная геометрия"[24]. Дано множество горизонтальных и вертикальных отрезков. Надо найти все точки пересечения отрезков.

Эти два примера объединяют то, что в обоих случаях поиск производится однократно. Это порождает свои особенности таких задач поиска. Как правило, данные в этих задачах не имеют сложной организации. Фиксация множества, в котором производится поиск, однозначно определяет множество найденных объектов. "Хорошесть" алгоритма поиска определяется при варьировании множества, в котором производится поиск.

В третьем понимании задачи поиска предполагается многократное обращение к одним и тем же данным, но возможно каждый раз с разными требованиями к искомым объектам, то есть с разными запросами на поиск. Такие задачи поиска обычно возникают в системах, использующих базы данных. Многократное использование порождает особую проблему — проблему специальной организации данных, направленной на последующее ускорение поиска. Процесс такой специальной организации данных, проводимый до того, как осуществляется поиск, называется предобработкой и часто может занимать очень большое время, которое затем окупается сторицей в результате многократности поиска. Простейшим примером предобработки является сортировка. Построение "хорошего" алгоритма поиска в этом случае сводится к нахождению хороших структур данных, то есть к осуществлению такой хорошей предобработки данных, которая обеспечила бы хорошую скорость поиска. "Хорошеть" алгоритма поиска в этом случае определяется варьированием запроса на поиск, например, как среднее время поиска на запросе.

Такие задачи поиска, возникающие в системах баз данных и предполагающие многократное обращение к одним и тем же данным, и являются объектом рассмотрения в данной работе.

Спецкурс "Оптимальный поиск в базах данных", на основе которого написана данная работа, читается на кафедре математической и программной защиты информации факультета защиты информации Российского государственного гуманитарного университета для специальности № 220600 "Организация и технология защиты информации". Целью спецкурса является описание подхода к исследованию сложностных характеристик алгоритмов поиска с помощью математического моделирования. В задачи спецкурса входит: ввести строгую математическую модель алгоритмов поиска, направленную на исследование среднего времени поиска; описать структуры данных, используемых в алгоритмах поиска; описать сверхбыстрые в "среднем" алгоритмы решения геометрических

задач поиска, относящихся к интенсивно развивающемуся в последние годы направлению, получившему название "Вычислительная геометрия" [24]. В отличие от спецкурса в данной книге практически не нашли отражения известные алгоритмы и структуры данных, которые можно найти, например, в классических учебниках Д. Кнута [23], Дж. Мартина [25], А. Ахо, Дж. Хопкрофта, Дж. Ульмана [3].

Приведем некоторые определения и обозначения, используемые в дальнейшем изложении.

Пусть M — некоторое конечное множество. Через $|M|$ обозначим число элементов во множестве M , называемое *мощностью* множества M .

Через $\{\overline{1, m}\}$ договоримся обозначать множество $\{1, 2, \dots, m\}$.

Некоторые оценки мы будем приводить с точностью до главного члена, поэтому введем обозначения, обычно принятые при описании асимптотических оценок.

Будем писать $\alpha(n) = \bar{o}(1)$, если $\lim_{n \rightarrow \infty} \alpha(n) = 0$.

Будем писать $A(n) = \bar{o}(B(n))$, если $A(n) = B(n) \cdot \bar{o}(1)$.

Скажем, что $A(n)$ асимптотически не превосходит $B(n)$ при $n \rightarrow \infty$ и обозначим $A \lesssim B$, если существует $\alpha(n) = \bar{o}(1)$ такое, что начиная с некоторого номера n_0 , $A(n) \leq (1 + \alpha(n)) \cdot B(n)$.

Если $A \lesssim B$ и $B \lesssim A$, то будем говорить, что A и B асимптотически равны при $n \rightarrow \infty$ и обозначать $A \sim B$.

Будем писать $A \asymp B$, если существует такая положительная константа c , что, начиная с некоторого номера n_0 , $A(n) \leq c \cdot B(n)$.

Если $A \lesssim B$ и $B \lesssim A$, то будем говорить, что A и B равны по порядку при $n \rightarrow \infty$ и обозначать $A \asymp B$ или $A = \underline{O}(B)$.

Через $\binom{n}{k}$ будем обозначать *число сочетаний из n элементов по k* .

Договоримся опускать значок основания для логарифмов по основанию 2, то есть будем использовать обозначение $\log x$ вместо $\log_2 x$.

Если r — действительное число, то через $[r]$ будем обозначать максимальное целое, не превышающее r , а через $]r[$ — минимальное целое, не меньшее, чем r .

В заключение автор хотел бы выразить благодарность В.Б.Кудрявцеву и А.С.Подколзину за помощь в работе, Э.А.Применко и А.С.Строгалову за ценные замечания, а также Российскому фонду фундаментальных исследований за частичную финансовую поддержку (грант 95-01-00597 "Исследование сложности алгоритмов поиска").

Глава 1

Информационные сети как математическая модель алгоритмов поиска информации

1.1 Понятие информационной сети

В настоящее время одним из актуальнейших направлений развития современной математической науки являются вопросы проектирования автоматизированных информационных систем. Причем широкое распространение получила концепция баз данных, согласно которой ядром информационной системы становятся данные, определенным образом организованные. Структуры организации данных при этом выбираются в соответствии со многими критериями, среди которых одним из основных является время поиска информации. И поэтому вполне понятно то внимание в литературе, которое проявляется к проблемам информационного поиска [2, 3, 23, 24, 25, 28, 29]. Среди этих работ хотелось бы выделить те, которые связаны с исследованием вычислительной сложности алгоритмов поиска информации. Основная масса работ в этом направлении связана с разработкой новых эффективных алгоритмов поиска, находящих многочисленные приложения в различных областях, таких, как машинное проектирование, машинная графика, библиотечно-информационные системы, робототехника, системы искусственного интеллекта и многих других [3, 23, 24, 27, 31, 35, 51, 57]. В этих работах оценивается сложность предлагаемых алгоритмов (чаще всего порядок сложности) и сравнивается со сложностью ранее разработанных алгоритмов. В ряде работ исповедуется другой подход, связанный с введением математических моделей вычислений, используемых главным образом для получения нижних оценок сложности вычислений [3, 5, 10, 34, 50, 58]. Среди этих моделей наиболее известной является так называемое алгебраическое дерево вычислений Бен-Ора [34]. Как разновидность алгебраического дерева вычислений можно рассматривать алгебраическое дерево решений порядка d [58]. В случае, когда d равно 1, получается линейное дерево решений, с использованием которого получены доказательства ряда нижних оценок сложности [48, 49, 50, 59].

В данной работе также предлагаются некие математические модели вычислений, но если модель Бен-Ора и другие предназначались для использования в довольно широком классе вычислительных процессов (см. [24]), то рассматриваемые в настоящей работе модели предназначаются для исследования алгоритмов поиска информации. Такая "узкая специализация" предлагаемых моделей позволяет надеяться на получение более интересных результатов, в частности более точных оценок сложности.

При разработке математической модели алгоритмов информационного поиска в данной работе используются принципы и аппарат теории управляющих систем и вводится новый класс управляющих систем, так называемых информационных сетей с переключателями (ИСП) (см. [13, 15, 16, 17]). Такое длинное название появилось потому, что исторически данный класс возник как обобщение более узкого класса информационных сетей [12]. Но в данной работе информационные сети будут вводиться как частный случай информационных сетей с переключателями. В свою очередь информационные сети появились как обобщение информационных деревьев, введенных автором в [5, 8].

Прежде чем вводить данный класс ИСП, формализуем понятие задачи информационного поиска (ЗИП). В работах [2, 3, 24, 28, 29] вводились различные формализации ЗИП, но мы дадим собственную формализацию, более удобную для использования в дальнейшем.

Пусть нам даны два множества Y и X . Первое множество Y является множеством объектов поиска. Из элементов этого множества составляются информационные массивы, в которых производится поиск нужных объектов. Элементы множества Y будем называть *записями*. Второе множество X назовем *множеством запросов*, а его элементы — *запросами*. Пусть на декартовом произведении $X \times Y$ задано бинарное отношение ρ , то есть задано некое подмножество $R \subseteq X \times Y$ и $x \rho y$, если $(x, y) \in R$. Отношение ρ будем называть *отношением поиска*. В содержательном смысле ρ описывает критерий семантического соответствия записи запросу, и мы будем говорить, что запись $y \in Y$ *удовлетворяет* запросу $x \in X$, если $x \rho y$.

Тройку $S = \langle X, Y, \rho \rangle$, где X — множество запросов, Y — множество записей, ρ — отношение поиска, заданное на $X \times Y$, будем называть *типом* или иногда более развернуто — *типом задач информационного поиска*.

Тройку $I = \langle X, V, \rho \rangle$, где X — множество запросов; V — некоторое конечное подмножество множества Y , в дальнейшем называемое *библиотекой*; ρ — отношение поиска, заданное на $X \times V$, будем называть *задачей информационного поиска* (ЗИП) типа $S = \langle X, Y, \rho \rangle$ (или ЗИП, принадлежащей типу S , и обозначать $I \in S$). Будем считать, что задача $I = \langle X, V, \rho \rangle$ 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 .

Для полной определенности отметим один факт. Всегда будем считать, что в библиотеке V все элементы различные, то есть нет повторяющихся элементов.

Если f — одноместный предикат, определенный на X , то есть $f : X \rightarrow \{0, 1\}$, 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 $y \in Y$.

Функцию $\chi_{y, \rho} : X \rightarrow \{0, 1\}$ такую, что $N_{\chi_{y, \rho}} = O(y, \rho)$, назовем *характеристической функцией* записи y .

Пусть F — множество символов одноместных предикатов, определенных на множестве X , называемое базовым множеством предикатов.

Пусть G — множество символов одноместных переключателей, определенных на множестве X . Под переключателями будем понимать функции, областью значений которых являются конечные подмножества натурального ряда.

Пару $\mathcal{F} = \langle F, G \rangle$ назовем базовым множеством.

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

предикат, определенный на X , такой, что

$$N_{\xi_g^n} = \{x \in X : g(x) = n\}.$$

Обозначим

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

где \mathbf{N} — множество натуральных чисел.

Пусть нам даны множество запросов X , множество записей Y и базовое множество $\mathcal{F} = \langle F, G \rangle$. Определение понятия ИСП, вводимое с помощью этих множеств можно, разбить на два этапа. На первом этапе раскрывается структурная (схемная) часть этого понятия, на втором — функциональная.

Определение ИСП с точки зрения ее структуры

Пусть нам дана ориентированная многополюсная сеть.

Выделим в ней один полюс и назовем его *корнем*, а остальные полюса назовем *листьями*.

Выделим в сети некоторые вершины и назовем их *точками переключения* (полюса могут быть точками переключения).

Если β — вершина сети, то через ψ_β обозначим *полустепень исхода* вершины β .

Каждой точке переключения β сопоставим некий символ из G , такой, что максимальное значение переключателя, соответствующего этому символу, не превышает ψ_β . Это соответствие назовем *нагрузкой* точек переключения.

Для каждой точки переключения β ребрам, из нее исходящим, поставим во взаимно однозначное соответствие числа из множества $\{\overline{1}, \overline{\psi_\beta}\}$. Эти ребра назовем *переключательными*, а это соответствие — *нагрузкой переключательных ребер*.

Ребра, не являющиеся переключательными, назовем *предикатными*.

Каждому предикатному ребру сети сопоставим некоторый символ из множества F . Это соответствие назовем *нагрузкой предикатных ребер*.

Сопоставим каждому листу сети некоторую запись из множества Y . Это соответствие назовем *нагрузкой листьев*.

Полученную нагруженную сеть назовем *информационной сетью с переключателями* над базовым множеством $\mathcal{F} = \langle F, G \rangle$.

Определение функционирования ИСП

Пусть нам дана ИСП U .

Последовательность ориентированных ребер сети

$$(\alpha_1, \alpha_2), (\alpha_2, \alpha_3), \dots, (\alpha_{m-1}, \alpha_m)$$

назовем *ориентированной цепью* от вершины α_1 к вершине α_m .

Если c — ребро сети, то через $[c]$ обозначим его нагрузку.

Проводимостью ребра (α, β) назовем предикат, равный $[(\alpha, \beta)]$, если ребро предикатное, и $\xi_g^{[(\alpha, \beta)]}$, если ребро переключательное, где g — переключатель, соответствующий вершине α .

Проводимостью ориентированной цепи назовем конъюнкцию проводимостей ребер цепи.

Если зафиксировать запрос x , то цепь, проводимость которой на запросе x равна 1, назовем *проводящей цепью* на запросе x .

В ИСП по аналогии с контактными схемами введем для каждой пары вершин α и β функцию проводимости $f_{\alpha\beta}$ от вершины α к вершине β следующим образом:

- если $\alpha = \beta$, то $f_{\alpha\beta}(x) \equiv 1$ ($x \in X$) ;
- если $\alpha \neq \beta$ и в ИСП не существует ориентированных цепей от α к β , то $f_{\alpha\beta}(x) \equiv 0$;
- если $\alpha \neq \beta$ и множество ориентированных цепей от α к β не пусто, то $f_{\alpha\beta}(x)$ равно дизъюнкции проводимостей всех ориентированных цепей от α к β .

Функцию проводимости от корня ИСП к некоторой вершине β ИСП назовем *функцией фильтра* вершины β и обозначим $\varphi_\beta(x)$.

Через $\mathcal{R}(U), \mathcal{P}(U), \mathcal{L}(U)$ (или просто $\mathcal{R}, \mathcal{P}, \mathcal{L}$) обозначим множества вершин, точек переключения и листьев сети U соответственно.

Пусть \mathcal{N} — некоторая подсеть (то есть произвольное подмножество вершин и ребер) ИСП U . Через $\langle \mathcal{N} \rangle$ обозначим множество записей, соответствующих листьям этой подсети (в частности, если α — некоторый лист сети U , то под $\langle \alpha \rangle$ будем понимать запись, соответствующую листу α).

Будем говорить, что ИСП U реализует функцию $\mathcal{J} : X \rightarrow 2^Y$, называемую *функцией ответа* сети U и определяемую соотношением:

$$\mathcal{J}(x) = \langle \{\alpha \in \mathcal{L}(U) : \varphi_\alpha(x) = 1\} \rangle.$$

Понятие ИСП полностью определено.

В случае, когда базовое множество переключателей G пусто, то есть в сетях нет переключателей, то ИСП называются *информационными сетями с дублированием листьев* (ИСД). Понятие ИСД хронологически предшествовало понятию ИСП и было впервые опубликовано в [13].

ИСД, различным листьям которой соответствуют различные записи, называется *информационной сетью* (ИС). Это понятие впервые введено в [10]. Более доступными изданиями являются [11, 12].

ИС, граф которой является деревом, а листья совпадают с висячими вершинами дерева, назовем *информационным деревом* (ИД).

Впервые понятие ИД было опубликовано в работах [5, 7, 6, 8, 9].

ИД удобны и интересны тем, что структуры данных, им соответствующие, практичны и их гораздо проще реализовать на ЭВМ.

1.2 Необходимое и достаточное условие допустимости информационных сетей

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

Скажем, что ИСП 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

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

ИСП U , разрешающую ЗИП I , будем также называть *допустимой* для задачи I .

Пусть U — некоторая ИСП, y — запись из V . Через $L_U(y)$ обозначим множество листьев сети U , которым соответствует запись y .

Справедлива следующая теорема.

Теорема 1 ИСП U разрешает ЗИП $I = \langle X, V, \rho \rangle$ тогда и только тогда, когда для любой записи $y \in V$, такой, что $O(y, \rho) \neq \emptyset$, справедливо $L_U(y) \neq \emptyset$ и $\bigvee_{\alpha \in L_U(y)} \varphi_\alpha = \chi_{y, \rho}$, а для любой записи $y \in V$, такой, что $O(y, \rho) = \emptyset$, справедливо либо $L_U(y) = \emptyset$, либо $\bigvee_{\alpha \in L_U(y)} \varphi_\alpha \equiv 0$.

Доказательство. Достаточность.

Возьмем произвольный запрос $x \in X$.

Возьмем произвольную запись $y \in V$.

Если $O(y, \rho) = \emptyset$ и, следовательно, $x \notin O(y, \rho)$, то по предположению либо $L_U(y) = \emptyset$, либо $\bigvee_{\alpha \in L_U(y)} \varphi_\alpha \equiv 0$. Откуда следует, что $y \notin \mathcal{J}(x)$.

Теперь рассмотрим случай, когда $O(y, \rho) \neq \emptyset$.

Если $x \rho y$, то $\chi_{y, \rho}(x) = 1$, и согласно предположению

$$\bigvee_{\alpha \in L_U(y)} \varphi_\alpha(x) = 1.$$

Откуда следует, что существует лист $\alpha \in \mathcal{L}(U)$ такой, что $\varphi_\alpha(x) = 1$. Следовательно, $y \in \mathcal{J}(x)$.

Если $x \notin O(y, \rho)$, то $\chi_{y, \rho}(x) = 0$, и согласно предположению $\bigvee_{\alpha \in L_U(y)} \varphi_\alpha(x) = 0$. Откуда следует, что $y \notin \mathcal{J}(x)$.

Таким образом, мы показали, что

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

и тем самым доказали достаточность.

Необходимость.

Возьмем произвольную запись $y \in V$.

Если $O(y, \rho) = \emptyset$ и, следовательно, для любого запроса $x \in X$ $x \notin O(y, \rho)$, значит, для любого $x \in X$ $y \notin \mathcal{J}(x)$. Откуда следует, что либо $L_U(y) = \emptyset$, либо $\bigvee_{\alpha \in L_U(y)} \varphi_\alpha \equiv 0$.

Теперь рассмотрим случай, когда $O(y, \rho) \neq \emptyset$.

Предположим, что для данной записи y не выполняются предположения теоремы, то есть существует такой запрос x , что

$$\bigvee_{\alpha \in L_U(y)} \varphi_\alpha(x) \neq \chi_{y, \rho}(x).$$

Но это означает, что y принадлежит в точности одному из множеств $\mathcal{J}(x)$ или $\{y \in V : x \rho y\}$.

Следовательно, при этом x

$$\mathcal{J}(x) \neq \{y \in V : x \rho y\},$$

и, значит, ИСП U не разрешает ЗИП I .

Тем самым теорема доказана.

По сути теорема 1 говорит, что если нам дана ЗИП $I = \langle X, V, \rho \rangle$ и мы хотим построить сеть, разрешающую эту ЗИП, мы должны построить многополюсник, который между корнем и остальными полюсами реализует как функции проводимости все функции $\chi_{y, \rho}$, где $y \in V$.

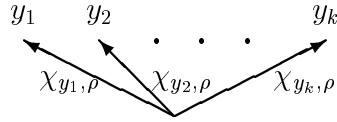


Рис. 1.1: Информационная сеть переборного алгоритма

1.3 К вопросу о полноте для информационных сетей

Если нам дана ЗИП $I = \langle X, V, \rho \rangle$ и базовое множество $\mathcal{F} = \langle F, G \rangle$, то возникает вопрос, а можно ли построить информационную сеть над базовым множеством \mathcal{F} , разрешающую эту задачу I ? Если для любой записи $y_i \in V = \{y_1, \dots, y_k\}$ $\chi_{y_i, \rho} \in F$, то ответ на этот вопрос положительный, и сеть, изображенная на рисунке 1.1, разрешает задачу I .

В данном разделе дается более полный ответ на этот вопрос.

Пусть нам дан тип $S = \langle X, Y, \rho \rangle$, где X — множество запросов, Y — множество записей, ρ — отношение поиска, заданное на $X \times Y$, и базовое множество $\mathcal{F} = \langle F, G \rangle$.

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 ИСП U над базовым множеством \mathcal{F} , разрешающая ЗИП I .

Справедлив следующий результат, относящийся к проблеме полноты для ИСП.

Теорема 2 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 $f_{ij} \in F \cup \hat{G}$.

Доказательство. Достаточность.

Пусть нам дана произвольная ЗИП $I = \langle X, V, \rho \rangle$, где $V = \{y_1, \dots, y_k\} \subseteq Y$. По предположению каждую из функций $\chi_{y_l, \rho}(x)$ ($l = \overline{1, k}$) можно представить формулой

$$\chi_{y_l, \rho}(x) = \bigvee_{i=1}^{n_l} \bigwedge_{j=1}^{m_{li}} f_{lij}(x),$$

где $f_{lij} \in F \cup \hat{G}$, $l = \overline{1, k}$.

ИСП, разрешающую ЗИП I , будем строить следующим образом.

Сначала возьмем $k + 1$ вершину и объявим одну из них корнем, а остальные объявим листьями и мысленно перенумеруем, начиная с 1 до k .

Затем для каждого $l \in \{\overline{1, k}\}$ проделаем следующее.

Припишем l -му листу (обозначим его α_l) запись y_l .

Если $O(y_l, \rho) \neq \emptyset$, то проведем из корня в лист α_l n_l ориентированных цепей, причем i -я цепь ($i = \overline{1, n_l}$) будет состоять из m_{li} ребер. Теперь для каждого i ($i \in \{\overline{1, n_l}\}$) проделаем следующее. Если $f_{lij} \in F$, то j -е ребро i -й цепи объявим предикатным и припишем ему предикат f_{lij} . Если $f_{lij} \in \hat{G}$ (то есть $f_{lij} = \xi_g^n$, где $g \in G$, а $n \in \mathbf{N}$), то вершину β , из которой исходит j -е ребро i -й цепи, объявим переключательной припишем ей переключатель g , j -му ребру i -й цепи припишем число n , из вершины β выпустим еще $r - 1$ ребро, где r — мощность области значений переключателя g , и сопоставим им взаимно однозначно числа из множества $\{\overline{1, r}\} \setminus \{n\}$.

Нетрудно видеть, что в этом случае

$$\varphi_{\alpha_l}(x) = \bigvee_{i=1}^{n_l} \bigwedge_{j=1}^{m_{li}} f_{lij}(x) = \chi_{y_l, \rho}(x).$$

Поскольку это условие выполняется для всех листьев, то согласно теореме 1 построенная ИСП разрешает ЗИП I .

Необходимость.

Пусть \mathcal{F} полно для отношения ρ . Возьмем произвольную запись $y \in Y$ такую, что $O(y, \rho) \neq \emptyset$. Пусть V — любая библиотека, содержащая запись y .

Так как \mathcal{F} полно, то существует ИСП U , разрешающая ЗИП $I = \langle X, V, \rho \rangle$. Рассмотрим множество $L_U(y)$ листьев сети U , которым соответствует запись y . Так как U разрешает задачу I , то согласно теореме 1

$$\bigvee_{\alpha \in L_U(y)} \varphi_\alpha(x) \equiv \chi_{y, \rho}(x).$$

А так как каждая из функций $\varphi_\alpha(x)$ есть функция проводимости от корня к листу α , а функция проводимости по определению есть дизъюнкция конъюнкций некоторых предикатов из $F \cup \hat{G}$, то необходимость доказана, что и доказывает теорему.

1.4 Понятие сложности информационных сетей

Каждой ИСП U можно сопоставить некий алгоритм. Предполагается, что этот алгоритм хранит в своей (внешней) памяти структуру ИСП U . Входными данными алгоритма является запрос. Выходными данными является множество записей.

Опишем этот алгоритм.

Пусть на вход алгоритма поступил запрос x . Работу алгоритма начинаем из корня сети U , объявляя его единственной текущей вершиной первого шага. Объявляем текущим шагом первый шаг. Просматриваем по очереди текущие вершины текущего шага. Если текущая вершина есть точка переключения, то вычисляем на запросе x переключатель, соответствующий данной вершине, и объявляем конец ребра, исходящего из текущей вершины, нагрузка которого равна значению переключателя, текущей вершиной следующего шага, если только эта вершина не была текущей на предыдущих шагах. Если текущая вершина не является точкой переключения, 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, объявляем текущими вершинами следующего шага, если только на предыдущих шагах эти вершины не объявлялись текущими. Затем переходим к следующему шагу, на котором данная процедура повторяется для всех текущих вершин очередного шага. Через некоторое количество шагов мы попадем во все вершины, функции фильтров которых равны 1 на запросе x . Если среди этих вершин есть листья, то записи, соответствующие этим листьям, включаем в выходные данные алгоритма. Остается заметить, что если ИСП разрешает задачу I , то множество, полученное на выходе алгоритма, будет содержать все те и только те записи библиотеки $\langle U \rangle$, которые удовлетворяют запросу x . Т.е. полученный алгоритм решает ЗИП $I = \langle X, V, \rho \rangle$, где $V = \langle U \rangle$, и, значит, является алгоритмом поиска.

Таким образом, ИСП как управляющая система может рассматриваться как модель алгоритма поиска, работающего над данными, организованными в структуру, определяемую структурой ИСП.

Определим понятие сложности ИСП на запросе.

Будем считать, что время вычисления любого переключателя из G примерно одинаково и характеризуется числом a , а время вычисления любого предиката из F — числом b .

Пусть нам дана некая ИСП U и произвольно взятый запрос $x \in X$. Пусть A — определенный ранее алгоритм, сопоставленный ИСП U .

Сложностью ИСП U на запросе x назовем число $T(U, x)$, равное количеству переключателей, вычисленных алгоритмом A при подаче на его вход запроса x , умноженное на a , плюс количество вычисленных предикатов, умноженное на b , то есть

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

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

Сложность ИСП можно вводить по-разному, например как максимальную сложность на запросе, как обычно и делается, но в данной работе мы введем понятие сложности ИСП как среднее значение сложности ИСП 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 $\mathbf{P}(X) = 1$.

В связи с тем, что мы ввели вероятностное пространство над множеством запросов, уточним понятие типа. А именно, под *типовом* будем понимать тройку $S = \langle X, Y, \rho \rangle$, считая, что множество запросов X рассматривается вместе со своим вероятностным пространством $\langle X, \sigma, \mathbf{P} \rangle$. В тех же случаях, когда мы хотим явно выделить рассматриваемое вероятностное пространство над X , мы будем представлять тип пятеркой $S = \langle X, Y, \rho, \sigma, \mathbf{P} \rangle$.

Справедлива следующая лемма.

Лемма 1 Если алгебра σ содержит все множества N_f , где $f \in F \cup \hat{G}$, то для любой ИСП U над базовым множеством $\mathcal{F} = \langle F, G \rangle$ функция $T(U, x)$ как функция от x является случайной величиной.

Доказательство. Нам необходимо доказать, что для любой ИСП U над базовым множеством \mathcal{F} и любого действительного числа r множество

$$\{x \in X : T(U, x) < r\} \in \sigma.$$

Покажем, что $(\beta \in \mathcal{R}(U)) \rightarrow (N_{\varphi_\beta} \in \sigma)$.

Пусть $\beta \in \mathcal{R}(U)$. Пусть \mathcal{C}_β — множество всех ориентированных цепей сети U , ведущих из корня в вершину β . Пусть C — некоторая цепь, а c — некоторое ребро. Через $\theta(c)$ обозначим проводимость ребра c .

Нетрудно видеть, что

$$\varphi_\beta = \bigvee_{C \in \mathcal{C}_\beta} \bigwedge_{c \in C} \theta(c).$$

Учитывая, что $N_{f \vee g} = N_f \cup N_g$, $N_{f \wedge g} = N_f \cap N_g$, имеем

$$N_{\varphi_\beta} = \bigcup_{C \in \mathcal{C}_\beta} \bigcap_{c \in C} N_{\theta(c)} \in \sigma.$$

Введем следующее обозначение:

$$\mathcal{M}_r = \{\mathcal{B} \subset \mathcal{R}(U) : |\{\mathcal{B} \cap \mathcal{P}\}| + \sum_{\beta \in \mathcal{B} \setminus \mathcal{P}} \psi_\beta < r\}.$$

Тогда, как нетрудно видеть,

$$\{x \in X : T(U, x) < r\} = \bigcup_{B \in \mathcal{M}_r} ((\bigcap_{\beta \in B} N_{\varphi_\beta}) \bigcap (\bigcap_{\beta \in \mathcal{R} \setminus B} (X \setminus N_{\varphi_\beta}))) \in \sigma.$$

Тем самым лемма доказана.

Далее всюду будем предполагать, что мы находимся в условиях леммы 1.

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

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

Если (β, α) — ребро ИСП, то *сложностью этого ребра* назовем число

- $b \cdot \mathbf{P}(N_{\varphi_\beta})$ — если (β, α) — предикатное ребро;
- $a \cdot \mathbf{P}(N_{\varphi_\beta})/\psi_\beta$ — если это ребро переключательное.

Если β — вершина ИСП, то число $\mathbf{P}(N_{\varphi_\beta})$ назовем *сложностью вершины* β .

Нетрудно показать, что сложность ИСП равна сумме сложностей ребер ИСП. В самом деле

$$\begin{aligned} T(U) &= \mathbf{M}_x T(U, x) = \int_X T(U, x) \mathbf{P}(dx) = \\ &= \int_X (b \cdot \sum_{\beta \in \mathcal{R} \setminus \mathcal{P}} \psi_\beta \cdot \varphi_\beta(x) + a \cdot \sum_{\beta \in \mathcal{P}} \varphi_\beta(x)) \mathbf{P}(dx) = \\ &= b \cdot \sum_{\beta \in \mathcal{R} \setminus \mathcal{P}} \psi_\beta \int_X \varphi_\beta(x) \mathbf{P}(dx) + a \cdot \sum_{\beta \in \mathcal{P}} \int_X \varphi_\beta(x) \mathbf{P}(dx) = \\ &= b \cdot \sum_{\beta \in \mathcal{R} \setminus \mathcal{P}} \psi_\beta \mathbf{P}(N_{\varphi_\beta}) + a \cdot \sum_{\beta \in \mathcal{P}} \mathbf{P}(N_{\varphi_\beta}). \end{aligned}$$

Далее всюду будем предполагать, что $a = b = 1$.

Пусть нам дана ИСП U .

Объемом $Q(U)$ ИСП U назовем число ребер в сети U .

В качестве примера мы можем подсчитать сложность сети U , изображенной на рисунке 1.1. Легко видеть, что $Q(U) = k$ и $T(U) = k$, то есть объем сети минимально возможный, а время максимальное. Это и не удивительно, так как сеть U соответствует переборному алгоритму поиска.

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

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

где $\mathcal{U}(I, \mathcal{F})$ — множество всех ИСП 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 существует такая ИСП $U \in \mathcal{U}(I, \mathcal{F})$, что $T(U) = T(I, \mathcal{F})$, то ИСП U будем называть *оптимальной* для ЗИП I .

Скажем, что некая вершина ИСП *достижима из корня* или просто *достижимая*, если функция фильтра этой вершины не равна тождественному нулю, в противном случае вершину называем *недостижимой*.

Скажем, что ребро ИСП *несущественное*, если выполняется хотя бы одно из следующих условий

- оно исходит из недостижимой вершины,
- оно является предикатным и входит в корень или в недостижимую вершину,
- оно является переключательным и число, приписанное этому ребру, больше максимального возможного значения переключателя, соответствующего началу этого ребра,
- начало и конец ребра совпадают.

В противном случае ребро называем *существенным*.

Легко заметить, что удаление несущественных ребер из сети не изменяет функционирования сети и не увеличивает ее сложность. Поэтому всегда в дальнейшем мы будем рассматривать ИСП с точностью до несущественных ребер, а точнее будем считать, что все ребра в ИСП — существенные.

Теорема 3 (о существовании оптимальных сетей) Пусть $S = \langle X, Y, \rho, \sigma, \mathbf{P} \rangle$ — такой тип, что множество запросов X конечно, σ — множество всех подмножеств X , \mathbf{P} — такая вероятностная мера, что для любого $B \subseteq X$ $\mathbf{P}(B) > 0$. Тогда, если базовое множество $\mathcal{F} = \langle F, G \rangle$ полно для типа S и множество $F \cup \hat{G}$ конечно, то для любой ЗИП I типа S существует оптимальная ИСП.

Доказательство. Возьмем произвольную ЗИП I типа S . Поскольку \mathcal{F} полно для типа S , то существует ИСП U_0 над базовым множеством \mathcal{F} , разрешающая ЗИП I. Очевидно, что

$$T(I, \mathcal{F}) = \inf\{T(U) : U \in \mathcal{U}(I, \mathcal{F})\} = \inf\{T(U) : U \in \mathcal{U}'\},$$

где $\mathcal{U}' = \{U \in \mathcal{U}(I, \mathcal{F}) : T(U) \leq T(U_0)\}$.

Для доказательства существования оптимальной сети для ЗИП I нам достаточно показать конечность множества \mathcal{U}' .

Пусть $|F \cup \hat{G}| = n$. Пусть M — множество всех отличных от тождественного нуля предикатов, полученных из предикатов множества $F \cup \hat{G}$ с помощью операций конъюнкции и дизъюнкции. Понятно, что $|M| \leq 2^{2^n}$. Пусть $\min_{f \in M} \mathbf{P}(N_f) = r$. Согласно условию теоремы $r > 0$. Пусть m — максимально возможное значение переключателей из G . Поскольку для любой ИСП над \mathcal{F} функции фильтров вершин, из которых исходят какие-либо ребра, принадлежат множеству M , то сложность любого предикатного ребра сети над \mathcal{F} не меньше, чем r , а сложность любого переключательного ребра не меньше, чем r/m . Отсюда в любой ИСП из множества \mathcal{U}' число ребер не больше, чем $T(U_0) \cdot m/r$. Поскольку из конечности множества $F \cup \hat{G}$ следует конечность числа различных нагрузок сетей, то, значит, множество \mathcal{U}' конечное.

Что и доказывает теорему.

1.5 Тривиальная нижняя оценка

Пусть нам даны произвольные множества запросов X , записей Y и отношение поиска ρ на $X \times Y$. Причем на множестве запросов задано вероятностное пространство $\langle X, \sigma, \mathbf{P} \rangle$. Следующий результат, называемый тривиальной нижней оценкой, справедлив для любой ЗИП при минимальных ограничениях. Смысл этого результата заключается в том, что время поиска не может быть меньше, чем время, необходимое на перечисление ответа.

Теорема 4 (тривиальная нижняя оценка) Пусть $I = \langle X, V, \rho \rangle$ — произвольная ЗИП, \mathcal{F} — базовое множество, удовлетворяющее условию леммы 1, такое, что множество $\mathcal{U}(I, \mathcal{F}) \neq \emptyset$, тогда

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

Доказательство. Возьмем произвольную ИСП U , разрешающую задачу I . Такая сеть существует, так как $\mathcal{U}(I, \mathcal{F}) \neq \emptyset$.

Возьмем произвольный запрос $x \in X$. Так как сеть U разрешает ЗИП I , то ответ на запрос x

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

Возьмем произвольную запись $y \in \mathcal{J}(x)$. Поскольку запись y попала в ответ, то, значит, в сети U существует некий лист α , которому приписана запись y и такой, что $\varphi_\alpha(x) = 1$. А так как $\varphi_\alpha(x) = 1$ и так как никакой лист не совпадает с корнем, то существует цепь, ведущая из корня в лист α , проводимость которой равна 1, и в этой цепи есть ребро, ведущее в α , с проводимостью 1. Это ребро назовем проводящим ребром записи y . Понятно, что разным записям из \mathcal{J} соответствуют разные проводящие ребра, так как эти ребра ведут в разные листья. Если проводящее ребро записи предикатное, предикат, приписанный проводящему ребру, обязательно был вычислен перед тем, как мы попали в лист. Если проводящее ребро записи переключательное, то обязательно был вычислен переключатель, приписанный вершине, из которой исходит проводящее ребро. Причем такие переключатели для разных записей из \mathcal{J} будут разными, так как только одно из переключательных ребер, исходящих из одной вершины, может иметь проводимость, равную 1. Таким образом каждой записи из \mathcal{J} можно сопоставить переключатель или предикат, вычисляемый непосредственно перед попаданием в соответствующий записи лист. Причем разным записям будут сопоставлены разные переключатели или предикаты. Отсюда следует, что

$$T(U, x) \geq |\mathcal{J}(x)|.$$

Следовательно,

$$\begin{aligned} T(U) &= \mathbf{M}_x T(U, x) \geq \mathbf{M}_x |\mathcal{J}(x)| = \\ &= \int_X |\mathcal{J}(x)| \mathbf{P}(dx) = \int_X |\{y \in V : x\rho y\}| \mathbf{P}(dx) = \\ &= \sum_{y \in V} \int_{O(y, \rho)} \mathbf{P}(dx) = \sum_{y \in V} \mathbf{P}(O(y, \rho)). \end{aligned}$$

А так как это неравенство выполняется для любой сети $U \in \mathcal{U}(I, \mathcal{F})$, то

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

что и требовалось доказать.

В случае когда базовое множество переключателей пусто, то есть в случае ИСД этот результат можно доказывать по-другому.

Если \mathcal{N} — некоторое множество ребер некоторой ИСД, то через $T(\mathcal{N})$ будем обозначать сумму сложностей ребер из \mathcal{N} .

Если U — некоторая ИСД, а β — некоторая вершина сети U , то через $\mathcal{N}_{U, \beta}$ будем обозначать множество ребер сети U , входящих в вершину β .

Лемма 2 Если $S = \langle X, Y, \rho, \sigma, \mathbf{P} \rangle$ — некоторый тип задач информационного поиска, где X — множество запросов, Y — множество записей, ρ — отношение поиска, заданное на $X \times Y$, σ — некоторая алгебра подмножеств множества X , \mathbf{P} — вероятностная мера на σ , и U — некоторая ИСД, разрешающая некоторую ЗИП $I = \langle X, V, \rho \rangle$ типа S , то для любой записи $y \in V$ справедливо

$$T\left(\bigcup_{\alpha \in L_U(y)} \mathcal{N}_{U,\alpha}\right) \geq \mathbf{P}(O(y, \rho)).$$

Доказательство. Возьмем произвольную запись $y \in V$. Через \mathcal{B} обозначим множество вершин, из которых исходят ребра, ведущие непосредственно в листья из $L_U(y)$, то есть \mathcal{B} — множество начал ребер из $\bigcup_{\alpha \in L_U(y)} \mathcal{N}_{U,\alpha}$.

Так как ИСД U разрешает ЗИП I , то согласно теореме 1

$$\bigcup_{\beta \in \mathcal{B}} N_{\varphi_\beta} \supseteq \bigcap_{\alpha \in L_U(y)} N_{\varphi_\alpha} = O(y, \rho).$$

Из-за аддитивности меры \mathbf{P}

$$T\left(\bigcup_{\alpha \in L_U(y)} \mathcal{N}_{U,\alpha}\right) \geq \sum_{\beta \in \mathcal{B}} \mathbf{P}(N_{\varphi_\beta}) \geq \mathbf{P}\left(\bigcup_{\beta \in \mathcal{B}} N_{\varphi_\beta}\right) \geq \mathbf{P}(O(y, \rho)).$$

Тем самым лемма доказана.

Доказательство теоремы 4 для ИСД.

Возьмем произвольную ИСД $U \in \mathcal{U}(I, \mathcal{F})$. Нетрудно заметить, что для любых $y, y' \in V$, таких, что $y \neq y'$, справедливо

$$\bigcup_{\alpha \in L_U(y)} \mathcal{N}_{U,\alpha} \bigcap \bigcup_{\alpha \in L_U(y')} \mathcal{N}_{U,\alpha} = \emptyset.$$

Тогда согласно лемме 2

$$T(U) \geq \sum_{y \in V} T\left(\bigcup_{\alpha \in L_U(y)} \mathcal{N}_{U,\alpha}\right) \geq \sum_{y \in V} \mathbf{P}(O(y, \rho))$$

для любой $U \in \mathcal{U}(I, \mathcal{F})$. Следовательно,

$$T(I, \mathcal{F}) = \inf_{U \in \mathcal{U}(I, \mathcal{F})} T(U) \geq \sum_{y \in V} \mathbf{P}(O(y, \rho)),$$

что и требовалось доказать.

Глава 2

Мгновенно решаемые задачи поиска

В данной главе исследуются некоторые задачи, которые являются одними из базисных в геометрических задачах поиска, получивших распространение в связи с развитием вычислительной геометрии, компьютерной графики и других компьютерных дисциплин. Описание и области использования этих задач можно найти, например, в [24, 27].

Согласно теореме 4 решение любой задачи информационного поиска (ЗИП) требует времени не меньшего, чем время, требуемое на перечисление ответа, которое тем самым является тривиальной нижней оценкой сложности ЗИП.

Поэтому естественный интерес представляет проблема нахождения таких алгоритмов решения ЗИП, которые по временной сложности несущественно отличаются от тривиальной нижней оценки, в частности, отличаются на константу, не зависящую от размерности задачи. Задачи поиска, для которых существуют такие алгоритмы, предлагается называть мгновенно решаемыми задачами поиска.

Итак, под *мгновенно решаемыми задачами поиска* понимаются такие ЗИП, которые могут быть решены в среднем за время, необходимое на перечисление ответа, плюс некая не зависящая от размерности задачи константа. Здесь под размерностью задачи понимается объем библиотеки, в которой производится поиск.

В данной главе приводятся пять задач, для которых показано, что они мгновенно решаемые. Это известные задачи:

- задача поиска с отношением поиска, являющимся отношением линейного квазипорядка;
- задача поиска идентичных объектов, которая состоит в поиске в информационном массиве объекта, идентичного объекту-запросу;
- задачи о близости, которые состоят в поиске во множестве, в котором задан линейный порядок, объекта, ближайшего к объекту-запросу справа или слева;
- n -мерная задача интервального поиска, которая состоит в поиске в конечном подмножестве n -мерного пространства всех тех точек, которые попадают в n -мерный параллелепипед-запрос, где $n \geq 1$;
- n -мерная 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, где $n \geq 1$.

Но если в задаче с отношением поиска, являющимся отношением линейного квазипорядка, алгоритм, обеспечивающий мгновенное решение, не требует дополнительной памяти, то мгновенного решения задач, исследуемых в данной разделе, иногда удается добиться только за счет больших затрат в объеме памяти. Но в тех ситуациях, когда время является очень дорогим, такие алгоритмы могут быть очень полезны.

Результаты данной главы были анонсированы в [10, 14, 52, 53] и опубликованы в [11, 16, 18].

2.1 Задачи поиска с отношением поиска, являющимся отношением линейного квазипорядка

Задача поиска, в которой отношение поиска является отношением линейного квазипорядка, наверное, одна из самых распространенных и вместе с тем самых простых задач поиска. Эта задача имеет полное и окончательное решение, которое мы здесь приведем. Этот результат был анонсирован в [10] и опубликован в [11]. Кстати, в этих работах впервые было введено понятие информационных сетей, то есть данная задача была пробным камнем для используемого здесь аппарата.

Если X — некоторое множество, то под *отношением линейного квазипорядка* \succeq^l на $X \times X$ будем понимать бинарное отношение, для любых $x, y, z \in X$ удовлетворяющее условиям

- рефлексивности $x \succeq^l x$;
- транзитивности $(x \succeq^l y) \& (y \succeq^l z) \rightarrow (x \succeq^l z)$;
- связности $(x \succeq^l y) \vee (y \succeq^l x)$.

Итак, мы будем рассматривать типы задач поиска, в которых в качестве отношения поиска выступает отношение линейного квазипорядка, а именно рассмотрим следующий тип:

$$S_{lin} = \langle X, X, \succeq^l \rangle,$$

где X — некоторое множество, \succeq^l — некоторое отношение линейного квазипорядка на $X \times X$.

Пусть $a \in X$, K_a — функция, действующая из X в $\{0, 1\}$, такая что $N_{K_a} = \{x \in X : x \succeq^l a\}$. Отметим, что $K_a(x)$ есть характеристическая функция записи a . Пусть $\mathcal{K} = \{K_a(x) : a \in X\}$.

Будем рассматривать следующее базовое множество $\mathcal{F}_0 = \langle \mathcal{K}, \emptyset \rangle$. Так как оно содержит характеристические функции всех записей, то $\mathcal{F}_0 = \langle \mathcal{K}, \emptyset \rangle$ полно для типа S_{lin} .

Теорема 5 Для любой ЗИП $I = \langle X, V, \succeq^l \rangle$ тунд S_{lin}

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

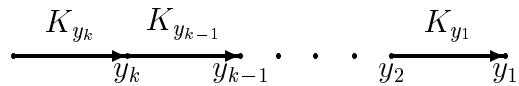


Рис. 2.1: Сеть U_0

Доказательство. Возьмем произвольную ЗИП $I = \langle X, V, \succeq \rangle^l$ типа S_{lin} . Пусть $V = \{y_1, y_2, \dots, y_k\}$, причем записи упорядочены так, что $y_1 \succeq y_2 \succeq \dots \succeq y_k$.

Верхняя оценка. Рассмотрим ИСД U_0 , изображенную на рисунке 2.1.

Обозначим лист, которому приписана запись y_i ($i = \overline{1, k}$), через α_i . Так как для любого $i \in \{\overline{1, k}\}$

$$\varphi_{\alpha_i} = \bigwedge_{j=i}^k K_{y_j} = K_{y_i},$$

то согласно теореме 1 сеть U_0 разрешает ЗИП I.

Нетрудно подсчитать, что

$$T(U_0) = 1 + \sum_{j=2}^k \mathbf{P}(O(y_j, \succeq)) = 1 + \sum_{y \in V} \mathbf{P}(O(y, \succeq)) - \min_{y \in V} \mathbf{P}(O(y, \succeq)).$$

Что и доказывает верхнюю оценку.

Нижняя оценка следует из теоремы 4.

Тем самым теорема полностью доказана

2.2 Поиск идентичных объектов

В данном разделе мы будем рассматривать задачу поиска идентичных объектов, которая в том или ином виде встречается во всех информационных системах и базах данных. Задача поиска идентичных объектов состоит в поиске в информационном массиве объекта, идентичного объекту-запросу. Опишем эту задачу формально.

Пусть нам дано множество X , на котором задано отношение линейного порядка \preceq , то есть такое бинарное отношение на $X \times X$, которое для любых $x, y, z \in X$ удовлетворяет условиям

- рефлексивности $x \preceq x$;
 - транзитивности $(x \preceq y) \& (y \preceq z) \rightarrow (x \preceq z)$;
 - антисимметричности $(x \preceq y) \& (y \preceq x) \rightarrow (x = y)$;
 - связности $(x \preceq y) \vee (y \preceq x)$.

Рассмотрим следующий тип задач поиска

$$S_{id} = \langle X, X, \rho_{id} \rangle,$$

где отношение поиска ρ_{id} есть отношение идентичности, то есть

$$x\rho_{id}y \iff x = y.$$

Тип S_{id} будем называть типом поиска идентичных объектов.

Пусть $\mathbf{N}_0 = \mathbf{N} \cup \{0\}$ — множество целых неотрицательных чисел.

Пусть

$$L_1(l) = \begin{cases} 0, & \text{если } l = 0 \\ \lfloor \log l \rfloor + 1, & \text{если } l = 1, 2, 3 \\ \log l + 2, & \text{если } l \geq 4 \end{cases} \quad (2.1)$$

функция, определенная на множестве \mathbf{N}_0 .

Докажем следующее вспомогательное утверждение.

Лемма 3 Пусть $L_1(l)$ — функция, определенная выше, пусть $k, m \in \mathbf{N}$, пусть

$$r_1(k, m) \stackrel{\text{def}}{=} \max \left\{ \sum_{i=1}^m L_1(l_i) : l_1 \in \mathbf{N}_0, \dots, l_m \in \mathbf{N}_0, \sum_{i=1}^m l_i = k \right\}.$$

Тогда

$$\begin{aligned} r_1(k, m) &= \left(k - \left\lfloor \frac{k}{m} \right\rfloor \cdot m \right) \cdot L_1 \left(\left\lfloor \frac{k}{m} \right\rfloor + 1 \right) + \\ &\quad + \left(m - k + \left\lfloor \frac{k}{m} \right\rfloor \cdot m \right) \cdot L_1 \left(\left\lfloor \frac{k}{m} \right\rfloor \right). \end{aligned}$$

Доказательство: Если доопределить функцию $L_1(l)$ на положительную полуось числовой прямой, например, следующим образом:

$$L_1(x) = \begin{cases} x, & \text{если } 0 \leq x \leq 4 \\ \log x + 2, & \text{если } x \geq 4 \end{cases},$$

то полученная функция будет непрерывной и вогнутой. И, вообще говоря, вогнутость этой функции объясняет результат леммы.

Более подробное доказательство будем вести от противного.

Предположим, что лемма неверна, то есть

$$\begin{aligned} r_1(k, m) &= \sum_{i=1}^m L_1(l'_i) > \left(k - \left\lfloor \frac{k}{m} \right\rfloor \cdot m \right) \cdot L_1 \left(\left\lfloor \frac{k}{m} \right\rfloor + 1 \right) + \\ &\quad + \left\lfloor \frac{k}{m} \right\rfloor \cdot m \cdot L_1 \left(\left\lfloor \frac{k}{m} \right\rfloor \right), \end{aligned} \quad (2.2)$$

и среди чисел $l'_i (i = \overline{1, m})$ существует 2 числа, разность которых не меньше двух.

Без ограничения общности можем считать, что $l'_1 - l'_2 \geq 2$.

Пусть

$$l''_1 = \left\lceil \frac{l'_1 + l'_2}{2} \right\rceil, \quad l''_2 = \left\lceil \frac{l'_1 + l'_2}{2} \right\rceil.$$

$$l''_1 + l''_2 = l'_1 + l'_2 \text{ и } l''_1 - l''_2 \leq 1.$$

Так как функция $L_1(x)$ вогнутая, то по свойству вогнутых функций

$$L_1(l''_1) + L_1(l''_2) \geq L_1(l'_1) + L_1(l'_2), \quad j = 1, 2. \quad (2.3)$$

Если в неравенстве (2.3) неравенство строгое, то получили противоречие, так как $\sum_{i=1}^m L_1(l'_i)$ — не максимальна; если же нет, то обозначим $l''_i = l'_i (i = \overline{3, m})$ и перейдем к рассмотрению суммы

$$\sum_{i=1}^m L_1(l''_i) = \sum_{i=1}^m L_1(l'_i) = r_1(k, m).$$

Если среди чисел l_i'' ($i = \overline{1, m}$) нет пары чисел, разность которых превышает 1, то получим противоречие с неравенством (2.2), если же есть, то опять проделаем операцию, описанную выше, и избавимся от такой пары, и так будем проделывать до тех пор, пока либо не получим строгое неравенство в неравенстве (2.3), либо не придем к разбиению $l_1^{(n)}, \dots, l_m^{(n)}$, такому, что $\sum_{i=1}^m L_1(l_i^{(n)}) = r_1(k, m)$, и все они отличаются не более чем на 1, и, тем самым, получим противоречие с неравенством (2.2), что доказывает лемму 3.

Пусть на X задано вероятностное пространство $\langle X, \sigma, \mathbf{P} \rangle$.

Пусть $g_m^1(x)$ — переключатель такой, что

$$g_m^1(x) = i, \text{ если } x \in X_i \ (i = \overline{1, m}), \quad (2.4)$$

где X_1, X_2, \dots, X_m — разбиение множества X (т.е.

$$X = X_1 \cup X_2 \cup \dots \cup X_m$$

и $X_i \cap X_j = \emptyset$, если $i \neq j$) такое, что

$$\mathbf{P}(X_i) \leq c/m \ (i = \overline{1, m}), \quad (2.5)$$

где $c = const$, не зависящая от m .

Так как

$$\mathbf{P}(X) = \sum_{i=1}^m \mathbf{P}(X_i) \leq c,$$

то отсюда сразу следует, что $c \geq 1$.

Пусть

$$g_a^2(x) = \begin{cases} 1, & \text{если } x \preceq a \\ 2, & \text{в противном случае} \end{cases}, \quad a \in X, \quad (2.6)$$

$$f_a(x) = \begin{cases} 0, & \text{если } x \neq a \\ 1, & \text{если } x = a \end{cases}, \quad a \in X. \quad (2.7)$$

Пусть

$$G_1 = \{g_m^1(x) : m \in \mathbf{N}\}, \quad (2.8)$$

$$G_2 = \{g_a^2(x) : a \in X\}, \quad (2.9)$$

$$F = \{f_a(x) : a \in X\}, \quad (2.10)$$

$$\mathcal{F} = \langle F, G_1 \cup G_2 \rangle. \quad (2.11)$$

Справедлива следующая теорема.

Теорема 6 Пусть $I = \langle X, V, \rho_{id} \rangle$ — задача поиска идентичных объектов, то есть задача туннели S_{id} , где $|V| = k$, \mathcal{F} — базовое множество, определяемое соотношениями (2.4)–(2.11), c — константа, определяемая соотношением (2.5), $L_1(l)$ — функция, определяемая соотношением (2.1). Тогда

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

В частности,

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

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

Доказательство:

Построим сеть U_m^0 , имеющую вид дерева, следующим образом.

Возьмем вершину β_0 и объявим ее корнем сети U_m^0 . Выпустим из β_0 m ребер, припишем им числа от 1 до m , объявим β_0 точкой переключения и припишем ей переключатель $g_m^1(x)$.

Пусть $V_i = X_i \cap V$, $l_i = |V_i|$, $i = \overline{1, m}$.

Конец ребра с номером i обозначим β_i .

Для всех таких i , что $V_i \neq \emptyset$, проделаем следующую процедуру. Выпустим из вершины β_i бинарное сбалансированное дерево с l_i концевыми вершинами, высоты $\lceil \log l_i \rceil$, все ребра которого ориентированы от корня к концевым вершинам. Если в этом дереве есть внутренняя вершина, из которой исходят ребра, ведущие одно в концевую вершину, а другое во внутреннюю (если такая вершина есть, то она только одна), то из концевой вершины, в которую ведет одно из этих ребер, выпустим одно ребро. Полученное дерево обозначим D_i . Объявим концевые вершины этого дерева листьями и сопоставим им слева направо в порядке возрастания элементы из V_i . (Говоря о дереве, мы подразумеваем некоторую его укладку, и направления "влево", "вправо" определяются уже на этой укладке.)

Пусть β — произвольная внутренняя вершина дерева D_i . Обозначим через V_β множество записей, соответствующих листьям ветви, растущей из β . Пусть β' — вершина, в которую ведет левое (если оно одно, то единственное) ребро из β . Пусть

$$y_\beta = \max_{y \in V_{\beta'}} y.$$

Объявим все внутренние вершины дерева D_i , из которых исходят ребра, ведущие во внутренние вершины, вершинами переключения и для каждой такой вершины β левому ребру, из нее выходящему, припишем 1, а правому — 2, а самой вершине припишем переключатель $g_{y_\beta}^2(x)$.

Все ребра дерева D_i , входящие в листья, объявим предикатными и припишем каждому такому ребру, ведущему в лист с записью y , предикат $f_y(x)$.

Полученную сеть с $|V| = k$ листьями обозначим U_m^0 . Это сеть над базовым множеством \mathcal{F} .

Покажем, что U_m^0 разрешает задачу $I = \langle X, V, \rho_{id} \rangle$.

Пусть $y \in V_i$ ($i \in \{\overline{1, m}\}$), т.е. лист α принадлежит дереву D_i . В него ведет единственная цепь, состоящая из не более чем $\lceil \log l_i \rceil + 1$ ребра, причем все эти ребра, кроме последнего, переключательные. Так как $g_m^2(y) = i$ и первое ребро в этой цепи есть i -е ребро, исходящее из корня, то проводимость первого ребра в цепи равна 1. Последнее ребро в цепи — предикатное с предикатом $f_y(x)$, и его проводимость равна $f_y(y) = 1$. Покажем, что проводимость остальных ребер цепи равна 1. Возьмем произвольно одно из этих ребер (β, β') . Если (β, β') — левое, исходящее из β , то $y \in V_{\beta'}$ и предикат, приписанный вершине β , $g_{y_\beta}^2(y) = 1$, так как

$$y \leq y_\beta = \max_{y' \in V_{\beta'}} y'.$$

Если (β, β') — правое ребро, исходящее из β , то $y > y_\beta$ и $g_{y_\beta}^2(y) = 2$, тем самым проводимость ребра (β, β') в обоих случаях равна 1.

Таким образом, мы показали, что $\varphi_\alpha(x) = f_y(x)$. Учитывая произвольность листа α и теорему 1, получим, что сеть U_m^0 решает задачу $I = \langle X, V, \rho_{id} \rangle$.

Подсчитаем объем сети U_m^0 .

Поскольку каждое из D_i есть дерево, в котором полустепень исхода любой вершины равна 2, кроме, быть может, одной вершины, то если в дереве D_i нет вершины с полустепенью исхода 1, то в D_i будет ровно $2 \cdot l_i - 2$ ребра, а если в дереве D_i есть вершина с

полустепенью исхода 1, то в D_i будет $2 \cdot l_i - 1$ ребро. Отсюда следует, что

$$Q(U_m^0) \leq m + \sum_{i=1}^m \max(0, 2 \cdot l_i - 1) \leq 2 \cdot k + m - 1.$$

Подсчитаем сложность полученной сети U_m^0 .

Рассмотрим произвольный запрос $x \in X_i$. При $l_i = 0$ дерево D_i пусто и $T(U_0, x) = 1$. Если $l_i > 0$, то, как мы показали выше, активные на этом запросе вершины сети U_0^m (вершина β активна на запросе, если $\varphi_\beta(x) = 1$) образуют единственную цепь, ведущую из корня к вершине с искомой записью, если искомая запись есть в библиотеке, или некую подцепь этой цепи, если искомой записи нет. Причем в этой цепи не более, чем $\lceil \log l_i \rceil$ переключательных вершин и одна вершина, из которой исходит не более двух ребер с предикатами. Тем самым

$$T(U_0^m) \leq 2 + \lceil \log l_i \rceil \leq 1 + L_1(l_i),$$

где $L_1(l)$ — функция, определяемая соотношением (2.1).

По определению и учитывая лемму 3,

$$\begin{aligned} T(U_0^m) &= \mathbf{M}_x T(U_0^m, x) = \int_X T(U_0^m, x) \mathbf{P}(dx) = \\ &= \sum_{i=1}^m \int_{X_i} T(U_0^m, x) \mathbf{P}(dx) \leq \sum_{i=1}^m (1 + L_1(l_i)) \cdot \mathbf{P}(X_i) = \\ &= 1 + \sum_{i=1}^m L_1(l_i) \cdot \mathbf{P}(X_i) \leq 1 + \frac{c}{m} \sum_{i=1}^m L_1(l_i) \leq \frac{c}{m} \sum_{i=1}^m L_1(l_i) \leq \\ &\leq 1 + r_1(k, m) \frac{c}{m} = \\ &= \frac{c}{m} \left(\left(k - \left[\frac{k}{m} \right] \cdot m \right) \cdot L_1 \left(\left[\frac{k}{m} \right] + 1 \right) + \right. \\ &\quad \left. + \left(m - k + \left[\frac{k}{m} \right] \cdot m \right) \cdot L_1 \left(\left[\frac{k}{m} \right] \right) \right) + 1. \end{aligned}$$

Возьмем $m = \lceil c \cdot k \rceil$. При этом

$$Q(U_0^m) \leq 2k + \lceil c \cdot k \rceil - 1 \leq (2 + c) \cdot k.$$

Так как $c \geq 1$, то $m \geq k$. Если $m = k$, то $k/m = 1$ и

$$r_1(m, k) = 0 \cdot L_1(2) + k \cdot L_1(1) = k \cdot L_1(1).$$

Если $m > k$, то $k/m = 0$ и

$$r_1(m, k) = k \cdot L_1(1) + (m - k) \cdot L_1(0) = k \cdot L_1(1).$$

Следовательно,

$$T(I, \mathcal{F}, (2 + c) \cdot k) \leq T(U_0^m) \leq 1 + \frac{c}{\lceil c \cdot k \rceil} k \cdot L_1(1) < 2.$$

Если взять $m = k \cdot \alpha(k)$, где $\alpha(k) \rightarrow \infty$ при $k \rightarrow \infty$ и $\alpha(k) > 1$ при любом k , то

$$T(I, \mathcal{F}) \leq 1 + \frac{c \cdot k}{k \cdot \alpha(k)} \lesssim 1.$$

С другой стороны, $T(I, \mathcal{F}) \geq 1$, так как из корня должно исходить хотя бы одно ребро, поскольку $k > 0$.

Тем самым теорема доказана.

Приведем два примера задачи поиска идентичных объектов.

Пример 1. Пусть $S_{id1} = \langle X, X, \rho_{id} \rangle$ — тип поиска идентичных объектов, где $X = [0, 1]$, причем задано вероятностное пространство $\langle X, \sigma, \mathbf{P} \rangle$ над X такое, что \mathbf{P} задается функцией плотностью вероятности $p(x)$.

Пусть

$$\begin{aligned} X_1 &= \{x \in X : 0 \leq x \leq 1/m\}. \\ X_i &= \{x \in X : \frac{i-1}{m} < x \leq \frac{i}{m}\}, \quad i = \overline{2, m}. \end{aligned} \tag{2.12}$$

Тогда переключатель $g_m^1(x) = \max(1,]x \cdot m[)$ удовлетворяет соотношению (2.4).

Если $p(x) \leq c = const$, то $\mathbf{P}(X_i) \leq c/m$, и мы находимся в условиях теоремы 6.

Приведем неформальное описание алгоритма поиска, приведенного в доказательстве теоремы 6 применительно к задаче

$$I = \langle X, V, \rho_{id} \rangle$$

типа S_{id1} , где $|V| = k$.

Разобьем отрезок $[0, 1]$ на m равных частей в соответствии с соотношениями (2.12).

Каждой части сопоставим подмножество множества V , состоящее из точек, принадлежащих этой части.

Теперь для произвольной точки-запроса x поиск точки из V , идентичной запросу, будем вести следующим образом.

Определим ту часть отрезка, которой принадлежит запрос x . Ее номер равен $\max(1,]x \cdot m[)$.

Теперь во множестве, сопоставленном найденной части, осуществим обычный дихотомический поиск.

Согласно лемме 3, наибольшее среднее время поиска будет в том случае, когда точки из множества V равномерно распределены по всем m частям отрезка.

Если в качестве m взять $m = k$, то случай, когда в каждую часть попадет по одной точке из V , будет иметь наибольшую сложность. А поскольку найти или не найти объект во множестве мощности 1 можно за 1 шаг, то в среднем за 2 шага (на первом мы определили номер нужной части) мы можем решить задачу поиска идентичных объектов.

Пример 2. Пусть $S_{id2} = \langle X, X, \rho_{id} \rangle$ — тип поиска идентичных объектов, где $X = \{1, \dots, N\}$.

Пусть $\langle X, \sigma, \mathbf{P} \rangle$ — вероятностное пространство над X , где σ — множество всех подмножеств, а вероятностная мера \mathbf{P} определяет равномерную вероятность на множестве запросов X , то есть для любого $x \in X$ $\mathbf{P}(x) = 1/N$.

Пусть задано $m \in \mathbf{N}$.

Пусть $r = N - m \cdot [N/m]$.

Пусть X_1, \dots, X_m — такое разбиение множества, что

$$\begin{aligned} X_i &= \{x \in X : 1 + (i-1) \cdot ([N/m] + 1) \leq x \leq i \cdot ([N/m] + 1)\}, \\ &\quad i = \overline{1, r}, \\ X_i &= \{x \in X : r \cdot ([N/m] + 1) + 1 + (i-1-r) \cdot [N/m] \leq x \leq \\ &\quad \leq r \cdot ([N/m] + 1) + (i-r) \cdot [N/m]\}, \quad i = \overline{r+1, m}, \end{aligned}$$

и $g_m^1(x) = i$, если $x \in X_i$, $i \in \{\overline{1, m}\}$. Тогда

$$\mathbf{P}(X_i) \leq ([N/m] + 1)/N < 2/m,$$

и мы опять находимся в условиях теоремы 6 при $c = 2$.

2.3 Задачи о близости

Задачи о близости, которые состоят в поиске во множестве, в котором задан линейный порядок, объекта, ближайшего к объекту-запросу, также очень распространены в информационных системах и как самостоятельные задачи, и как задачи, возникающие при декомпозиции более сложных задач поиска. В частности, мы тоже будем использовать задачи о близости при решении многомерной задачи интервального поиска и задачи о доминировании.

Если X — множество запросов, Y — множество записей, 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 . Пусть множество запросов $X = Y$. Пусть $V = \{y_1, \dots, y_k\} \subseteq Y$.

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

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

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.

Пусть

$$L_2(l) = \begin{cases} l, & \text{если } 0 \leq l \leq 3 \\ \log(l+1) + 1, & \text{если } l \geq 3 \end{cases}. \quad (2.13)$$

Лемма 4 Пусть $L_2(l)$ — функция, определенная выше, пусть $k, m \in \mathbf{N}$, пусть

$$r_2(k, m) \stackrel{\text{def}}{=} \max \left\{ \sum_{i=1}^m L_2(l_i) : l_1 \in \mathbf{N}_0, \dots, l_m \in \mathbf{N}_0, \sum_{i=1}^m l_i = k \right\}.$$

Тогда

$$\begin{aligned} r_2(k, m) &= \left(k - \left[\frac{k}{m} \right] \cdot m \right) \cdot L_2 \left(\left[\frac{k}{m} \right] + 1 \right) + \\ &\quad + \left(m - k + \left[\frac{k}{m} \right] \cdot m \right) \cdot L_2 \left(\left[\frac{k}{m} \right] \right). \end{aligned}$$

Доказательство: Если доопределить функцию $L_2(l)$ на положительную полуось числовой прямой, например следующим образом:

$$L_2(x) = \begin{cases} x, & \text{если } 0 \leq x \leq 3 \\ \log(x+1) + 1, & \text{если } x \geq 3 \end{cases},$$

то полученная функция будет непрерывной и вогнутой. Дальнейшее доказательство полностью аналогично доказательству леммы 3, так как в нем использовалось только свойство непрерывности и вогнутости функции $L_1(x)$.

Тем самым лемма 4 доказана.

Пусть базовое множество

$$\mathcal{F} = \langle \emptyset, G_1 \cup G_2 \rangle, \quad (2.14)$$

где G_1 и G_2 определяются соотношениями (2.4), (2.6), (2.8), (2.9).

Справедлива следующая теорема.

Теорема 7 Пусть $I = \langle X, V, \rho_{near1} \rangle$ — первая задача о близости, где $|V| = k$, \mathcal{F} — базовое множество, определяемое соотношениями (2.4)–(2.9), (2.14), c — константа, определяемая соотношением (2.5), $L_2(l)$ — функция, определяемая соотношением (2.13). Тогда

$$\begin{aligned} 1 < T(I, \mathcal{F}, 2 \cdot k + m) &\leq \frac{c}{m} \left(\left(k - \left[\frac{k}{m} \right] \cdot m \right) \cdot L_2 \left(\left[\frac{k}{m} \right] + 1 \right) + \right. \\ &\quad \left. + \left(m - k + \left[\frac{k}{m} \right] \cdot m \right) \cdot L_2 \left(\left[\frac{k}{m} \right] \right) \right) + 1. \end{aligned}$$

В частности,

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

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

Доказательство: Построим сеть U_m^1 по аналогии с сетью U_m^0 из доказательства теоремы 6.

Возьмем вершину β_0 и объявим ее корнем сети. Выпустим из β_0 m ребер, припишем им числа от 1 до m , объявим β_0 точкой переключения и припишем ей переключатель $g_m^1(x)$.

Пусть $V_i = X_i \cap V$, $l_i = |V_i|$, $i = \overline{1, m}$.

Конец ребра с номером i обозначим β_i .

Для всех i таких, что $V_i \neq \emptyset$, проделаем следующую процедуру. Выпустим из вершины β_i бинарное сбалансированное дерево D_i с l_i+1 концевыми вершинами и высоты $\lceil \log(l_i+1) \rceil$.

Объявим все концевые вершины этого дерева D_i , кроме последней (самой правой), листьями и сопоставим им слева направо в порядке возрастания элементы из V_i .

Для произвольной внутренней вершины β дерева D_i введем понятия V_β и y_β , определяемые так же, как и в сети U_m^0 .

Объявим все внутренние вершины дерева D_i вершинами переключения и для каждой внутренней вершины β левому ребру, из нее выходящему, припишем 1, а правому - 2, а самой вершине припишем переключатель $g_{y_\beta}^2(x)$.

Пусть $i \in \{\overline{1, m}\}$. Обозначим $j(i)$ такой номер, что $j(i) > i$, $|V_{j(i)}| > 0$ и не существует $j' : |V_{j'}| > 0$ и $j' > i$ и $j' < j(i)$, то есть $j(i)$ — индекс ближайшего сверху непустого множества $V_{j(i)}$. Если такого множества нет, то $j(i) = 0$.

Теперь для каждого дерева D_i самую правую концевую вершину дерева отождествим с самым левым листом дерева $D_{j(i)}$, если $j(i) \neq 0$.

Для каждого такого i , что $l_i = 0$, вершину β_i отождествим с самым левым листом дерева $D_{j(i)}$, если $j(i) \neq 0$.

Полученная сеть и будет сетью U_m^1 .

Доказательство того, что сеть U_m^1 разрешает первую задачу о близости $I = \langle X, V, \rho_{near1} \rangle$, аналогично доказательству того, что сеть U_m^0 разрешает задачу поиска идентичных объектов.

Подсчитаем объем сети U_m^1 .

Поскольку каждое из D_i есть дерево, в котором полустепень исхода любой вершины равна 2, то в D_i будет ровно $2 \cdot (l_i + 1) - 2$ ребра. Отсюда следует, что

$$Q(U_m^1) = m + \sum_{i=1}^m (2(l_i + 1) - 2) = m + 2 \sum_{i=1}^m l_i = m + 2k.$$

Подсчитаем сложность сети U_m^1 .

Рассмотрим произвольный запрос $x \in X$. При $l_i = 0$

$$T(U_m^1, x) = 1.$$

Если $l_i > 0$, то активные на запросе x вершины сети U_m^1 образуют единственную цепь, ведущую из корня к вершине с искомой записью или же к самой правой вершине последнего дерева D_i . И в этой цепочке не более $1 + \log(l_i + 1)$ переключательных вершин и, значит,

$$T(U_m^1, x) \leq 1 + \log(l_i + 1).$$

Отсюда следует, что $T(U_m^1, x) \leq 1 + L_2(l_i)$.

Тогда как следствие леммы 4 и по аналогии с теоремой 6 доказывается утверждение теоремы 7.

Что и требовалось доказать.

Аналогично решается задача и когда надо найти ближайшую к запросу слева запись в библиотеке, т.е. когда ρ_{near2} определяется соотношением

$$x \rho_{near2} y \iff (y \in V) \& (y \preceq x) \& (\neg(\exists y')((y' \in V) \& (y' \preceq x) \& (y \prec y'))).$$

ЗоБ $I = \langle X, V, \rho_{near2} \rangle$ с таким отношением поиска назовем второй задачей о близости.

Примеры, приведенные в разделе 2.2, с таким же успехом могут служить примерами рассмотренных задач о близости.

2.4 Многомерная задача интервального поиска

Многомерная задача интервального поиска состоит в поиске в конечном подмножестве n -мерного пространства всех тех точек, которые попадают в n -мерный параллелепипед-запрос. Опишем тип задач поиска, который соответствует n -мерной задаче интервального поиска.

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

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

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:

$$(u_1, v_1, \dots, u_n, v_n) \rho_{intn} (y_1, \dots, y_n) \iff u_i \leq y_i \leq v_i, i = \overline{1, n}. \quad (2.15)$$

Тогда тип

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

назовем типом многомерного интервального поиска.

Пусть $V = \{\tilde{y}_1, \dots, \tilde{y}_k\}$ — конечная библиотека, элементы которой принадлежат Y_{intn} .

Пусть $S \subseteq V$, и $1 \leq i_1 < \dots < i_l \leq n$.

Обозначим через

$$\Pi^{i_1, \dots, i_l}(S) = \{(y_{i_1}, \dots, y_{i_l}) : (y_1, \dots, y_n) \in S\} \quad (2.16)$$

проекцию множества S на компоненты i_1, \dots, i_l .

Обозначим

$$W^i = \{(y', y'') : y', y'' \in \Pi^i(V) \text{ и } y' \leq y''\}, \quad i = \overline{1, n}. \quad (2.17)$$

Обозначим

$$\begin{aligned} Y^i &= \{(y_1^1, y_2^1, \dots, y_1^i, y_2^i) : y_1^j, y_2^j \in \Pi^j(V) \text{ и } y_1^j \leq y_2^j, j = \overline{1, i}\}, \\ i &= \overline{1, n}, \end{aligned} \quad (2.18)$$

Для каждой пары $(y', y'') \in W^i$ определим множество

$$S_{y', y''}^i = \{\tilde{y} = (y_1, \dots, y_n) \in V : y' \leq y_i \leq y''\}. \quad (2.19)$$

Пусть $\tilde{y}^i = (y_1^1, y_2^1, \dots, y_1^i, y_2^i) \in Y^i$ ($i = \overline{0, n-1}$). Здесь и далее под значком \tilde{y}^0 будем понимать пустое место или отсутствие значка, так, например, запись $p_{\tilde{y}^0}^1$ будем понимать как p^1 , а запись $M^1(\tilde{y}^0)$ — как M^1 . Обозначим

$$V^1 = V, \quad (2.20)$$

$$V^i(\tilde{y}^{i-1}) = \bigcap_{j=1}^{i-1} S_{y_1^j, y_2^j}^j, \quad i = \overline{2, n}, \quad (2.21)$$

$$M^i(\tilde{y}^{i-1}) = \Pi^i(V^i(\tilde{y}^{i-1})), \quad i = \overline{1, n}, \quad (2.22)$$

$$M_y^i(\tilde{y}^{i-1}) = \{y' \in M^i(\tilde{y}^{i-1}) : y' \geq y\}, \quad i = \overline{1, n}. \quad (2.23)$$

Будем считать, что каждое из множеств $M^i(\tilde{y}^{i-1})$ и $M_y^i(\tilde{y}^{i-1})$ ($i = \overline{1, n}$) упорядочено по возрастанию.

Пусть M — некоторое конечное, упорядоченное по возрастанию множество точек из отрезка $[0, 1]$. Пусть $y \in M$. Пусть y' — точка, предшествующая точке y в M , если y — не первая точка в M , и $y' = 0$, если y — первая точка в M . Обозначим

$$Z_y^1(M) = \begin{cases} [y', y], & \text{если } y \text{ — первая точка в } M, \\ (y', y], & \text{в противном случае.} \end{cases} \quad (2.24)$$

Пусть y'' — следующая за y точка в M , если y — не последняя в M , и $y'' = 1$, если y — последняя точка в M . Обозначим

$$Z_y^2(M) = \begin{cases} [y, y''], & \text{если } y \text{ — последняя точка в } M, \\ [y, y''), & \text{в противном случае.} \end{cases} \quad (2.25)$$

Обозначим

$$\tilde{x} = (u_1, v_1, \dots, u_n, v_n), \quad \tilde{z}_i = (u_i, v_i), \quad (2.26)$$

$$X_1 = \{(u, v) : 0 \leq u \leq v \leq 1\}, \quad (2.27)$$

$$X'_{y,z}(M) = \{(u, v) \in X_1 : u \in Z_y^1(M), v \in Z_z^2(M)\}, \quad (2.28)$$

где M — некоторое произвольное конечное, упорядоченное по возрастанию множество точек из отрезка $[0, 1]$,

$$\begin{aligned} p_i(u_1, v_1, \dots, u_i, v_i) &= \underbrace{\int_{X_1} \cdots \int_{X_1}}_{n-i} p(\tilde{x}) d\tilde{z}_{i+1} \cdots d\tilde{z}_n, \\ i &= \overline{1, n-1}, \end{aligned} \quad (2.29)$$

$$p_n(\tilde{x}) = p(\tilde{x}). \quad (2.30)$$

Пусть $\tilde{y}^{i-1} = (y_1^1, y_2^1, \dots, y_1^{i-1}, y_2^{i-1}) \in Y^{i-1}$, $y_i \in M^i(\tilde{y}^{i-1})$, где $i = \overline{1, n}$, тогда обозначим

$$p^1(u_1, v_1) = p_1(u_1, v_1), \quad (2.31)$$

$$\begin{aligned} p_{\tilde{y}^{i-1}}^i(u_i, v_i) &= \int_{X'_{y_1^{i-1}, y_2^{i-1}(M^{i-1}(\tilde{y}^{i-2}))}} d\tilde{z}_{i-1} \cdots \\ &\quad \left. \cdots \int_{X'_{y_1^1, y_2^1}(M^1)} p_i(u_1, v_1, \dots, u_i, v_i) d\tilde{z}_1 \right/ \\ &\quad \left/ \int_{X'_{y_1^{i-1}, y_2^{i-1}(M^{i-1}(\tilde{y}^{i-2}))}} d\tilde{z}_{i-1} \cdots \right. \\ &\quad \left. \cdots \int_{X'_{y_1^1, y_2^1}(M^1)} p_{i-1}(u_1, v_1, \dots, u_{i-1}, v_{i-1}) d\tilde{z}_1, \right. \\ i &= \overline{2, n}, \end{aligned} \quad (2.32)$$

$$p_{\tilde{y}^{i-1}}^{i,1}(u_i) = \int_{u_i}^1 p_{\tilde{y}^{i-1}}^i(u_i, v_i) dv_i, \quad i = \overline{1, n-1}, \quad (2.33)$$

$$\begin{aligned} p_{\tilde{y}^{i-1}, y_i}^{i,2}(v_i) &= \int_{Z_{y_i}^1(M^i(\tilde{y}^{i-1}))} p_{\tilde{y}^{i-1}}^i(u_i, v_i) du_i \Big/ \\ &\quad \left/ \int_{Z_{y_i}^1(M^i(\tilde{y}^{i-1}))} du_i \int_{u_i}^1 p_{\tilde{y}^{i-1}}^i(u_i, v_i) dv_i, \right. \\ i &= \overline{1, n-1}. \end{aligned} \quad (2.34)$$

Скажем, что функция плотности вероятности $p(\tilde{x})$, определяющая вероятностную меру \mathbf{P} , обладает свойством специальной ограниченности с константой c для библиотеки V , если для любого $i \in \{\overline{1, n-1}\}$, любого $\tilde{y}^{i-1} = (y_1^1, y_2^1, \dots, y_1^{i-1}, y_2^{i-1}) \in Y^{i-1}$ и любого $y_i \in M^i(\tilde{y}^{i-1})$ выполняются следующие условия:

$$p_{\tilde{y}^{i-1}}^{i,1}(u_i) \leq c,$$

$$(1 - y'_i) \cdot p_{\tilde{y}^{i-1}, y_i}^{i,2}(v_i) \leq c$$

и

$$p_{\tilde{y}^{n-1}}^n(u_n, v_n) \leq c,$$

где y'_i — левый конец отрезка $Z_{y_i}^1(M^i(\tilde{y}^{i-1}))$.

Очевидно, что $c \geq 1$.

Пусть

$$\begin{aligned} G_1 &= \{ g_{i,m}^1(u_1, v_1, \dots, u_n, v_n) = \\ &= \max(1,]u_i \cdot m[) : i \in \{\overline{1, n}\}, m \in \mathbf{N} \}, \end{aligned} \quad (2.35)$$

$$\begin{aligned} G_2 &= \{ g_{i,m}^2(u_1, v_1, \dots, u_n, v_n) = \\ &= \max(1,]v_i \cdot m[) : i \in \{\overline{1, n-1}\}, m \in \mathbf{N} \}, \end{aligned} \quad (2.36)$$

$$\begin{aligned} G_3 &= \{ g_{i,a}^3(u_1, v_1, \dots, u_n, v_n) = \\ &= \begin{cases} 1, & \text{если } u_i \leq a \\ 2, & \text{если } u_i > a \end{cases} : i \in \{\overline{1, n}\}, a \in [0, 1] \}, \end{aligned} \quad (2.37)$$

$$\begin{aligned} G_4 &= \{ g_{i,a}^4(u_1, v_1, \dots, u_n, v_n) = \\ &= \begin{cases} 1, & \text{если } v_i \leq a \\ 2, & \text{если } v_i > a \end{cases} : i \in \{\overline{1, n-1}\}, a \in [0, 1] \}. \end{aligned} \quad (2.38)$$

Обозначим

$$M_{a,b} = \{ \tilde{x} = (u_1, v_1, \dots, u_n, v_n) \in X_{intn} : u_n \leq b, v_n \geq a \}.$$

Пусть

$$F_1 = \{ f_{a,b} : N_{f_{a,b}} = M_{a,b}, 0 \leq a \leq b \leq 1 \}, \quad (2.39)$$

$$F_2 = \{ \neg f_{0,a} : a \in [0, 1], f_{0,a} \in F_1 \}, \quad (2.40)$$

$$\begin{aligned} G_5 &= \{ g_a^5(u_1, v_1, \dots, u_n, v_n) = \\ &= \begin{cases} 1, & \text{если } u_n \leq v_n < u_n + a \\ 2, & \text{в противном случае} \end{cases} : a \in [0, 1] \}. \end{aligned} \quad (2.41)$$

Пусть

$$\mathcal{F} = \langle F_1 \cup F_2, G_1 \cup G_2 \cup G_3 \cup G_4 \cup G_5 \rangle. \quad (2.42)$$

Справедлива следующая теорема.

Теорема 8 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, определяемое соотношениями (2.35)–(2.42), $n \geq 1$. Пусть

$$R(I) = \sum_{\tilde{y} \in V} \mathbf{P}(O(\tilde{y}, \rho_{intn})).$$

Тогда, если функция плотности вероятности $p(\tilde{x})$, определяющая вероятностную меру \mathbf{P} , обладает свойством специальной ограниченности с константой c для библиотеки V , то

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

Доказательство теоремы 8 проведем в два этапа. Сначала рассмотрим одномерный случай, т.е. когда $n = 1$, а затем многомерную задачу ($n > 1$).

2.4.1 Одномерный случай

В одномерном случае в качестве базового множества можно взять множество

$$\mathcal{F} = \langle F_1 \cup F_2, G_1 \cup G_3 \cup G_5 \rangle,$$

т.е. множества G_2 и G_4 нужны только для реализации многомерной задачи.

В одномерном случае свойство специальной ограниченности превращается в обычную ограниченность функции плотности вероятности $p(\tilde{x})$.

В одномерном случае $Y_{intn} = [0, 1]$, а библиотека V есть конечное множество точек из отрезка $[0, 1]$. Пусть $V = \{y_1, y_2, \dots, y_k\}$, причем $y_1 \leq y_2 \leq \dots \leq y_k$, т.е. V — множество, упорядоченное в порядке неубывания своих элементов.

Нижняя оценка следует из теоремы 4.

Построим ИСП, на которой достигается верхняя оценка одномерной задачи интервального поиска.

Возьмем точку, объявим ее корнем сети и обозначим β_0 . Выпустим из корня два ребра, одно из которых будем считать левым, а другое правым. Конец левого ребра обозначим β_1 , а конец второго — β_2 .

Пусть m — некоторый параметр, значение которого определим позже.

Припишем корню переключатель $g_{1/m}^5(u, v)$ из множества G_5 . Напомним, что $n = 1$, поэтому мы опустим индексы при переменных. Левому ребру припишем 1, а правому 2.

Выпустим из вершины β_1 дерево D , аналогичное деревьям D_i из доказательства теоремы 6. А именно, дерево D будем строить следующим образом.

Сначала возьмем бинарное сбалансированное дерево с k концевыми вершинами (напомним, что $k = |V|$), высоты $\lceil \log k \rceil$, все ребра которого ориентированы от корня к концевым вершинам. Если в этом дереве есть внутренняя вершина, из которой исходят ребра, ведущие одно в концевую вершину, а другое во внутреннюю (если такая вершина есть, то она только одна), то из концевой вершины, в которую ведет одно из этих ребер, выпустим одно ребро. Полученное дерево обозначим D .

Объявим концевые вершины этого дерева листьями и занумеруем их слева направо. Напомним, что, говоря о дереве, мы подразумеваем некоторую его укладку, и направления "влево", "вправо" определяются уже на этой укладке. Обозначим i -й лист через α_i и сопоставим ему запись y_i .

Пусть β — произвольная внутренняя вершина дерева D . Так же, как и в теореме 6, обозначим через V_β множество записей, соответствующих листьям ветви, растущей из β . Пусть β' — вершина, в которую ведет левое (если оно одно, то единственное) ребро из β . Пусть

$$y_\beta = \max_{y \in V_{\beta'}} y.$$

Объявим все внутренние вершины дерева D , из которых исходят ребра, ведущие во внутренние вершины, вершинами переключения и для каждой такой вершины β левому ребру, из нее выходящему, припишем 1, а правому — 2, а самой вершине припишем переключатель $g_{y_\beta}^3(x)$ из множества G_3 .

Все ребра дерева D , входящие в листья, объявим предикатными и припишем каждому такому ребру, ведущему в лист с записью y , предикат $f_{y,y}(x)$ из множества F_1 .

Для удобства дальнейшего изложения назовем множество переключательных ребер дерева D переключательной частью дерева D , а множество предикатных ребер — предикатной частью дерева D .

Теперь из каждого листа α_i ($i = \overline{1, k - 1}$) выпустим ребро, ведущее в лист α_{i+1} , и припишем ему предикат $f_{y_{i+1}, y_{i+1}} \in F_1$.

Это множество из $(k - 1)$ -го ребра назовем правосторонней концевой цепью.

Теперь из вершины β_2 (напомним, что это вершина, в которую ведет правое ребро, исходящее из корня) выпустим $m - 1$ ребро, припишем им числа от 1 до $m - 1$, объявим β_2 точкой переключения и припишем ей переключатель $g_{1,m}^1 \in G_1$ (напомним, что m — введенный ранее параметр). Отметим, что из β_2 мы выпустили именно $m - 1$ ребро, хотя переключатель $g_{1,m}^1$ может принимать m значений.

Конец ребра, исходящего из вершины β_2 и имеющего номер i , обозначим β_i^l .

Введем множество номеров $S = \{s_1, \dots, s_{m-1}\}$ такое, что s_i — номер записи из V такой, что y_{s_i} — ближайшая слева запись к точке i/m ($i = \overline{1, m-1}$), а если такой записи не существует, то $s_i = 0$.

Возьмем k новых точек, объявим их листьями и обозначим $\alpha'_1, \dots, \alpha'_k$. Каждому листу α_i ($i = \overline{1, k}$) припишем запись y_i (тем самым каждая запись y_i будет приписана двум листьям — α_i и α'_i). Из каждого листа α'_i ($i = \overline{2, k}$) выпустим ребро, ведущее в лист α'_{i-1} , и припишем ему предикат $f_{y_{i-1}, y_{i-1}} \in F_1$.

Это множество из $(k - 1)$ -го ребра назовем левосторонней концевой цепью.

Теперь для каждой вершины β_i^l проделаем следующие действия. Если $s_i \neq 0$, то из β_i^l выпустим ребро, ведущее в лист α'_{s_i} , и припишем ему предикат $f_{y_{s_i}, y_{s_i}} \in F_1$. Если $s_i < y_k$, то из β_i^l выпустим ребро, ведущее в лист α_{s_i+1} , и припишем ему предикат $f_{y_{s_i+1}, y_{s_i+1}} \in F_1$.

Это множество ребер, исходящих из вершин β_i^l ($i = \overline{1, m-1}$), назовем правой предикатной частью.

Полученную таким образом ИСП обозначим через U_0 .

Покажем, что сеть U_0 решает одномерную задачу интервального поиска $I = \langle X_{intn}, V, \rho_{intn} \rangle$.

Каждая запись $y_i \in V$ соответствует ровно двум листьям сети U_0 , т.е. $L_{U_0}(y_i) = \{\alpha_i, \alpha'_i\}$. Так как $O(y, \rho_{intn}) = N_{f_{y,y}}$ и так как в каждый лист α_i и α'_i ведут только ребра, которым приписаны предикаты f_{y_i, y_i} , то

$$N_{\varphi_{\alpha_i} \vee \varphi_{\alpha'_i}} \subseteq O(y_i, \rho_{intn}).$$

Тогда согласно теореме 1 достаточно показать, что для любого $y_i \in V$

$$N_{\varphi_{\alpha_i} \vee \varphi_{\alpha'_i}} \supseteq O(y_i, \rho_{intn}),$$

или, что то же самое, показать, что для $\forall x \in O(y_i, \rho_{intn})$ либо

$$\varphi_{\alpha_i}(x) = 1,$$

либо

$$\varphi_{\alpha'_i}(x) = 1,$$

то есть из корня либо в лист α_i , либо в лист α'_i существует проводящая цепь.

Обозначим $A_a = \{x = (u, v) : u \leq v \leq u + a\}$.

Возьмем произвольную запись $y_i \in V$.

Рассмотрим сначала случай, когда $x = (u, v) \in A_{1/m} \cap O(y_i, \rho_{intn})$. Это означает, что $v - u < 1/m$ и $u \leq y_i \leq v$.

Покажем, что из корня в лист α_i существует проводящая цепь.

В рассматриваемом случае $g_{1/m}^5(x) = 1$, и проводимость ребра (β_0, β_1) будет равна 1.

Отметим, что проводимость ребра (β_0, β_2) будет равна 0.

По аналогии с теоремами 6 и 7 нетрудно заметить, что в дереве D (растущем из вершины β_1) существует проводящая цепь, ведущая из β_1 в такой лист α_j , что запись y_j ,

ближайшая справа к u точка из V , принадлежащая отрезку $[u, v]$ (такая точка существует, так как $y_i \in [u, v]$). Таким образом, справедливо $u \leq y_j \leq y_i \leq v$. Отсюда легко видеть, что часть правосторонней концевой цепи, ведущая из y_j в y_i , является проводящей.

Таким образом, мы показали наличие проводящей на запросе x цепи, ведущей из корня в лист α_i .

Отметим также, что $\varphi_{\alpha'_i}(x) = 0$, так как в лист α'_i можно попасть только через ребро (β_0, β_2) , проводимость которого на запросе x , как мы уже отмечали, равна 0.

Рассмотрим теперь случай, когда

$$x = (u, v) \in (X \setminus A_{1/m}) \cap O(y_i, \rho_{intn}),$$

то есть $v - u \geq 1/m$ и $u \leq y_i \leq v$.

В этом случае $g_{1/m}^5(x) = 2$ и проводимость ребра (β_0, β_1) будет равна 0, а ребра (β_0, β_2) равна 1.

Пусть $j \in \{\overline{1, m-1}\}$ — такой номер, что j/m — ближайшая справа к u точка. Легко видеть, что $g_{1,m}^1(x) = j$.

Так как $v - u \geq 1/m$, то точка j/m принадлежит отрезку $[u, v]$.

Рассмотрим два случая.

1) $y_i \leq j/m$.

Тогда $u \leq y_i \leq y_{s_j} \leq v$, так как y_{s_j} — ближайшая слева к j/m запись из библиотеки V . Отсюда следует, что проводимость ребра, ведущего из β'_j в лист α'_{s_j} равна 1. Осталось заметить, что проводимость части левосторонней концевой цепи, ведущей из α'_{s_j} в α'_i , также будет равна 1, так как $0 \leq y_i \leq y_{s_j} \leq v$.

Отметим также, что в этом случае $\varphi_{\alpha_i}(x) = 0$, так как в лучшем случае мы попадем в вершины правосторонней концевой цепи в точке α_{s_j+1} , которая в этой цепи находится после точки α_i .

2) $y_i > j/m$.

Тогда $u \leq y_{s_j+1} \leq y_i \leq v$. Отсюда следует, что проводимость ребра, ведущего из β'_j в лист α_{s_j+1} , равна 1, и проводимость части правосторонней концевой цепи, ведущей из α_{s_j+1} в α'_i , также равна 1 на запросе x .

Аналогично предыдущему случаю $\varphi_{\alpha'_i}(x) = 0$.

Тем самым мы показали, что для $\forall y_i \in V$ и $\forall x \in X : x \rho_{intn} y_i$ в сети U_0 существует проводящая на запросе x цепь, ведущая из корня в какой-либо (но ровно в один) из листьев α_i и α'_i .

Что и доказывает, что сеть U_0 решает задачу I .

Подсчитаем теперь сложность сети U_0 .

Рассмотрим сначала произвольный запрос $x \in A_{1/m}$.

В этом случае

$$T(U_0, x) \leq 1 + (\lceil \log k \rceil - 1) + 2 + |\mathcal{J}(x)|.$$

Здесь первое слагаемое соответствует вычислению переключателя $g_{1/m}^5$ в вершине β_0 . Второе слагаемое дают переключатели, входящие в единственную проводящую цепь, пролегающую через переключательную часть дерева D . Третье слагаемое соответствует вычислению одного или двух предикатов, соответствующих предикатным ребрам из предикатной части дерева D , растущим из вершины, в которую ведет проводящая цепь. Четвертое слагаемое соответствует вычислению предикатов, соответствующих ребрам, исходящим из листьев, записи которых входят в ответ (по одному на каждую запись).

Рассмотрим теперь случай, когда $x \in X \setminus A_{1/m}$.

Тогда

$$T(U_0, x) \leq 1 + 1 + 2 + |\mathcal{J}(x)|.$$

Здесь первое слагаемое соответствует вычислению переключателя $g_{1/m}^5$, второе — переключателя $g_{1,m}^1$ из вершины β_2 , третье — вычислению одного или двух предикатов, приписанных ребрам, исходящим из той вершины β'_j , в которую ведет проводящее ребро из β_2 . И, наконец, четвертое слагаемое, как и ранее, соответствует вычислению предикатов, соответствующих ребрам, исходящим из листьев, записи которых входят в ответ. Как мы показали ранее, для любой записи y_i , вошедшей в ответ, ровно у одного из листьев α_i и α'_i функция фильтра будет равна 1, и, следовательно, каждой записи соответствует ровно одно ребро и соответственно ровно один вычисленный предикат.

Теперь мы можем подсчитать сложность сети U_0 .

$$\begin{aligned} T(U_0) &= \mathbf{M}_x T(U_0, x) = \mathbf{P}(A_{1/m}) \cdot (2 + \lceil \log k \rceil) + \\ &\quad + \mathbf{P}(X \setminus A_{1/m}) \cdot 4 + \mathbf{M}_x |\mathcal{J}(x)| \leq \\ &\leq \mathbf{P}(A_{1/m}) \cdot (\lceil \log k \rceil - 1) + 4 + \sum_{y \in V} \mathbf{P}(O(y, \rho_{intn})) \leq \\ &\leq c \cdot (\lceil \log k \rceil - 1) \left(\frac{2}{m} - \frac{1}{m^2} \right) + 4 + \sum_{y \in V} \mathbf{P}(O(y, \rho_{intn})) \leq \\ &\leq \frac{2c \cdot (\lceil \log k \rceil - 1)}{m} + 4 + \sum_{y \in V} \mathbf{P}(O(y, \rho_{intn})). \end{aligned}$$

Используемое здесь равенство

$$\mathbf{M}_x |\mathcal{J}(x)| = \sum_{y \in V} \mathbf{P}(O(y, \rho_{intn}))$$

доказывается простой сменой порядка суммирования.

Подсчитаем объем сети U_0 :

$$Q(U_0) \leq 2 + (2k - 1) + (k - 1) + (k - 1) + m + 2m.$$

Здесь первое слагаемое соответствует ребрам, исходящим из β_0 . Второе слагаемое есть количество ребер в дереве D . Третье и четвертое слагаемые соответствуют ребрам из правосторонней и левосторонней концевых цепочек. Пятое слагаемое — это ребра, исходящие из вершины β_2 . И, наконец, шестое слагаемое не меньше, чем число ребер, исходящих из вершин β'_i ($i = \overline{1, m}$).

Возьмем в качестве параметра $m = 2c \lceil \log k \rceil$ и получим

$$T(U_0) \leq 5 + \sum_{y \in V} \mathbf{P}(O(y, \rho_{intn})),$$

$$Q(U_0) \leq 4k - 1 + 6c \lceil \log k \rceil,$$

что и доказывает утверждение теоремы 8 в случае $n = 1$.

Дадим неформальное описание алгоритма, приведенного выше.

Пусть нам дано множество $V = \{y_1, \dots, y_k\}$, в котором мы должны производить поиск. Сначала упорядочим его в порядке возрастания. Если известна оценка сверху c функции плотности вероятности появления запросов, то в качестве параметра m возьмем $m = 2c \lceil \log k \rceil$, если же c неизвестна, то вместо нее можно взять любое число, например, $c = 2$. Затем для V построим множество номеров $S = \{s_1, \dots, s_{m-1}\}$, описанное выше. Отметим, что оно строится только однажды. 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 дихотомическим 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 $\log k$ действий.

Если $v - u \geq 1/m$, то с помощью функции $g_{1,m}^1$ получаем номер j точки j/m , попадающей в интервал $[u, v]$. Теперь, начиная с записи с номером s_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 $s_j + 1$, 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 $g_{1,m}^1$, делаем 1 лишнее действие, идя справа налево, и 1 лишнее действие, идя слева направо).

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.

И, наконец, заметим, что данный алгоритм требует дополнительную память порядка $\log k$, чтобы хранить множество S .

2.4.2 Многомерный случай

Приведем доказательство теоремы 8 при $n > 1$.

В данном пункте мы воспользуемся для нашей библиотеки $V = \{\tilde{y}_1, \dots, \tilde{y}_k\}$ понятиями и обозначениями, определенными в соотношениях (2.16)–(2.34).

Решать задачу будем методом снижения размерности. Покажем, как можно перейти от решения n -мерной задачи интервального поиска к решению $(n - 1)$ -мерной задачи.

Пусть $\tilde{x} = (u_1, v_1, \dots, u_n, v_n) \in X_{intn}$ — произвольный запрос.

Сначала найдем такую пару (y', y'') из Y^1 , что y' — ближайшая справа к u_1 точка из множества M^1 (напомним, что M^1 определяется соотношением (2.22) и является проекцией V на первую координату), а y'' — ближайшая слева к v_1 точка из M_y^1 , определяемого соотношением (2.23). Если такой пары нет, то ответ на запрос \tilde{x} пуст, если же есть, то мы можем перейти к решению для запроса $\tilde{x}' = (u_2, v_2, \dots, u_n, v_n)$ $(n - 1)$ -мерной задачи интервального поиска в множестве $\prod_{y_2, \dots, y_n} (V^2(y', y''))$, так как множество $V^2(y', y'')$, определяемое соотношением (2.21), содержит все точки $\tilde{y} = (y_1, \dots, y_n) \in V$ такие, что $u_1 \leq y_1 \leq v_1$.

Таким способом мы за $n - 1$ шаг придем к одномерной задаче интервального поиска, решение которой мы описали ранее.

Опишем ИСП, которая решает задачу I описанным выше методом.

Пусть $y \in M^1$, обозначим

$$X^1 = X_{intn}, \quad X_y^1 = \{\tilde{x} \in X^1 : u_1 \in Z_y^1(M^1)\}. \quad (2.43)$$

Пусть $\tilde{y}^{i-1} = (y_1^1, y_2^1, \dots, y_1^{i-1}, y_2^{i-1}) \in Y^{i-1}$, $y_i \in M^i(\tilde{y}^{i-1})$, где $i = \overline{1, n}$, тогда обозначим

$$X^i(\tilde{y}^{i-1}) = \{\tilde{x} \in X_{y_1^{i-1}}^{i-1}(\tilde{y}^{i-2}) : v_{i-1} \in Z_{y_2^{i-1}}^2(M^{i-1}(\tilde{y}^{i-2}))\}, \quad i = \overline{2, n}. \quad (2.44)$$

$$X_{y_i}^i(\tilde{y}^{i-1}) = \{\tilde{x} \in X^i(\tilde{y}^{i-1}) : u_i \in Z_{y_i}^1(M^i(\tilde{y}^{i-1}))\}, \quad i = \overline{2, n}. \quad (2.45)$$

Рассмотрим следующую задачу поиска: для произвольного запроса $\tilde{x} = (u_1, v_1, \dots, u_n, v_n) \in X_{in}$ найти в множестве M^1 , задаваемого соотношением (2.22), точку, ближайшую справа к точке u_1 . Эта задача (обозначим ее I^1) соответствует первой задаче о близости, в которой в качестве библиотеки взято множество M^1 , в качестве множества запросов взята проекция множества X^1 на ось u_1 , то есть отрезок $[0, 1]$, а вероятностная мера вероятностного пространства над множеством запросов определяется функцией плотности вероятности $p^{1,1}(u_1)$, задаваемой соотношением (2.33) и являющейся не чем иным, как функцией плотности вероятности координаты u_1 запросов, принадлежащих множеству X^1 .

Пусть m — параметр, определяющий мощность разбиения множества запросов. Если разбить множество запросов, то есть отрезок $[0, 1]$, на m равных частей, как в примере 1 из пункта о поиске идентичных объектов, то вероятность каждой части будет не больше, чем c/m , так как в силу специальной ограниченности $p^{1,1}(u_1) \leq c$. То есть мы находимся в условиях теоремы 7.

Поэтому построим ИСП U_m^1 , решающую первую задачу о близости так, как было описано в доказательстве теоремы 7, где в качестве параметра m возьмем $m = \lceil c \cdot |M^1| \rceil$. Тогда $Q(U_m^1) \leq 2k + \lceil c \cdot k \rceil$ и $T(U_m^1) < 2$.

Заменим в сети U_m^1 переключатель g_m^1 на $g_{1,m}^1$, а все переключатели вида $g_{y_\beta}^2$ на g_{1,y_β}^3 , поскольку мы все же хотим, чтобы U_m^1 решала задачу I^1 .

Возьмем произвольный лист α сети U_m^1 .

Пусть ему соответствует точка $y \in M^1$.

Мы попадем в лист α при всех запросах $\tilde{x} = (u_1, v_1, \dots, u_n, v_n)$, у которых точка y — ближайшая справа в M^1 к u_1 . Легко видеть, что это запросы из множества X_y^1 , задаваемого соотношением (2.43), то есть $\varphi_\alpha(x) = 1$ для любого $x \in X_y^1$.

Теперь рассмотрим следующую задачу поиска: для произвольного запроса $\tilde{x} = (u_1, v_1, \dots, u_n, v_n)$ найти в множестве M_y^1 , задаваемом соотношением (2.23), точку, ближайшую слева к точке v_1 . Эта задача (обозначим ее I_y^1) соответствует второй задаче о близости, в которой в качестве библиотеки взято множество M_y^1 , в качестве множества запросов взята проекция множества X_y^1 на ось v_1 , то есть отрезок $[y', 1]$, где y' — левый конец отрезка $Z_y^1(M^1)$, задаваемого соотношением (2.24), а вероятностная мера вероятностного пространства над множеством запросов определяется функцией плотности вероятности $p_y^{1,2}(v_1)$, задаваемой соотношением (2.34) и являющейся не чем иным, как функцией плотности вероятности координаты v_1 запросов, принадлежащих множеству X_y^1 .

Так как в силу специальной ограниченности

$$p_y^{1,2}(v_1) \leq c/(1 - y'),$$

то если разбить множество запросов на m равных частей, вероятность каждой части будет не больше, чем c/m . То есть мы можем воспользоваться алгоритмом, описанным в теореме 7.

Построим сеть $U_m^{1,y}$, решающую задачу I_y^1 . Для этого сначала построим сеть, решающую вторую задачу о близости для множества M_y^1 (отметим, что $M_y^1 \neq \emptyset$), где в качестве параметра m возьмем $m = \lceil c \cdot |M_y^1| \rceil$. Затем заменим в сети $U_m^{1,y}$ переключатель g_m^1 на $g_{1,m}^1$, а все переключатели вида $g_{y_\beta}^2$ на g_{1,y_β}^4 , поскольку сеть $U_m^{1,y}$ должна решать задачу I_y^1 .

Легко видеть, что

$$Q(U_m^{1,y}) = 2 \cdot |M_y^1| + \lceil c \cdot |M_y^1| \rceil \leq |M_y^1| \cdot (2 + c) + 1$$

и

$$T(U_m^{1,y}) < 2.$$

Объявим α внутренней вершиной и уберем приписанную ей точку y .

Теперь отождествим корень сети $U_m^{1,y}$ с вершиной α , т.е. сеть $U_m^{1,y}$ теперь будет расти из α , причем α не будет полюсом полученной сети.

И, наконец, для каждого листа α' сети $U_m^{1,y}$ заменим приписанную листу α' точку y' на пару (y, y') . Эту пару будем воспринимать как обозначение множества $V^2(y, y')$, заданного соотношением (2.21).

Проделаем такую операцию для каждого листа сети U_m^1 .

Полученную сеть обозначим U_1 .

Она имеет не более $k(k+1)/2$ листьев, которым взаимно однозначно сопоставлены пары из Y^1 . Эта сеть позволяет для любого запроса $\tilde{x} = (u_1, v_1, \dots, u_n, v_n) \in X$ находить пару $(y', y'') \in Y^1$ такую, что y' — ближайшая справа к u_1 точка из M^1 , а y'' — ближайшая слева к v_1 точка из $M_{y'}^1$, если, конечно, такая пара существует, т.е. по сути позволяет находить множество точек $V^2(y', y'') \subseteq V$, которые удовлетворяют первой паре неравенств из (2.15).

Так как в сети U_m^1 только 1 активный путь, то

$$T(U^1) \leq T(U_m^1) + \max_y T(U_m^{1,y}) \leq 2 + 2 = 4.$$

Легко видеть, что

$$Q(U^1) \leq (2+c)k + 1 + k + (2+c) \sum_{y \in V^1} |M_y^1| = k(k+3)(2+c)/2 + k + 1.$$

Опишем достаточно подробно и следующий шаг построения сети, решающей многомерную задачу интервального поиска, предполагая, что $n \geq 3$ (при $n = 2$ этот шаг не потребуется).

Рассмотрим произвольный лист α сети U^1 .

Пусть ему соответствует пара $(y, z) \in Y^1$.

Отметим, что $V^2(y, z)$ не пусто.

Теперь будем строить сеть, которая для множества $V^2(y, z)$ будет решать $(n-1)$ -мерную задачу интервального поиска, и выпустим эту сеть из вершины α .

Заметим, что только запросы из множества $X^2(y, z)$, определяемого соотношением (2.44), попадают в вершину α , то есть

$$\varphi_\alpha(x) = 1 \iff x \in X^2(y, z).$$

Поэтому рассмотрим следующую задачу поиска: для произвольного запроса $\tilde{x} = (u_1, v_1, \dots, u_n, v_n) \in X^2(y, z)$ найти в множестве $M^2(y, z)$, являющемся проекцией множества $V^2(y, z)$ на вторую координату и задаваемого соотношением (2.22), точку, ближайшую справа к точке u_2 . Эта задача (обозначим ее $I^2(y, z)$) соответствует первой задаче о близости, в которой в качестве библиотеки взято множество $M^2(y, z)$, в качестве множества запросов взята проекция множества $X^2(y, z)$ на ось u_2 , то есть отрезок $[0, 1]$, а вероятностная мера вероятностного пространства над множеством запросов определяется функцией плотности вероятности $p_{(y,z)}^{2,1}(u_1)$, задаваемой соотношением (2.33) и являющейся не чем иным, как функцией плотности вероятности координаты u_2 запросов, принадлежащих множеству $X^2(y, z)$.

Так как в силу специальной ограниченности

$$p_{(y,z)}^{2,1}(u_1) \leq c,$$

то если разбить множество запросов на m равных частей, вероятность каждой части будет не больше, чем c/m . То есть мы можем воспользоваться алгоритмом, описанным в теореме 7.

Построим сеть $U_m^{2,(y,z)}$, решающую задачу $I^2(y,z)$. Для этого сначала построим сеть, решающую вторую задачу о близости для множества $M^2(y,z)$ (отметим, что $M^2(y,z) \neq \emptyset$), где в качестве параметра m возьмем $m =]c \cdot |M^2(y,z)|[$. Затем заменим в сети $U_m^{2,(y,z)}$ переключатель g_m^1 на $g_{2,m}^1$, а все переключатели вида $g_{y_\beta}^2$ на g_{2,y_β}^3 , поскольку сеть $U_m^{2,(y,z)}$ должна решать задачу $I^2(y,z)$.

Легко видеть, что

$$Q(U_m^{2,(y,z)}) = 2 \cdot |M^2(y,z)| +]c \cdot |M^2(y,z)|[\leq |M^2(y,z)| \cdot (2 + c) + 1$$

и

$$T(U_m^{2,(y,z)}) < 2.$$

Возьмем произвольный лист α' сети $Q(U_m^{2,(y,z)})$.

Пусть ему соответствует точка $y' \in M^2(y,z)$.

Мы попадем в лист α' при всех запросах $\tilde{x} = (u_1, v_1, \dots, u_n, v_n)$, у которых точка y' — ближайшая справа в $M^2(y,z)$ к u_2 . Легко видеть, что это запросы из множества $X_{y'}^2(y,z)$, задаваемого соотношением (2.45), то есть

$$\varphi_{\alpha'}(x) = 1 \iff x \in X_{y'}^2(y,z).$$

Теперь рассмотрим следующую задачу поиска: для произвольного запроса $\tilde{x} = (u_1, v_1, \dots, u_n, v_n)$ найти в множестве $M_{y'}^2(y,z)$, задаваемом соотношением (2.23), точку, ближайшую слева к точке v_2 . Эта задача (обозначим ее $I_{y'}^2(y,z)$) соответствует второй задаче о близости, в которой в качестве библиотеки взято множество $M_{y'}^2(y,z)$, в качестве множества запросов взята проекция множества $X_{y'}^2(y,z)$ на ось v_2 , то есть отрезок $[y''', 1]$, где y''' — левый конец отрезка $Z_{y'}^1(M^2(y,z))$, задаваемого соотношением (2.24), а вероятностная мера вероятностного пространства над множеством запросов определяется функцией плотности вероятности $p_{(y,z),y'}^{2,2}(v_2)$, задаваемой соотношением (2.34) и являющейся не чем иным, как функцией плотности вероятности координаты v_2 запросов, принадлежащих множеству $X_{y'}^2(y,z)$.

Так как в силу специальной ограниченности

$$p_{(y,z),y'}^{2,2}(v_2) \leq c/(1 - y'''),$$

то если разбить множество запросов на t равных частей, вероятность каждой части будет не больше, чем c/t . То есть мы можем воспользоваться алгоритмом, описанным в теореме 7.

Построим сеть $U_m^{2,(y,z),y'}$, решающую задачу $I_{y'}^2(y,z)$. Для этого сначала построим сеть, решающую вторую задачу о близости для множества $M_{y'}^2(y,z)$ (отметим, что $M_{y'}^2(y,z) \neq \emptyset$), где в качестве параметра m возьмем $m =]c \cdot |M_{y'}^2(y,z)|[$. Затем заменим в сети $U_m^{2,(y,z),y'}$ переключатель g_m^1 на $g_{2,m}^1$, а все переключатели вида $g_{y_\beta}^2$ на g_{2,y_β}^4 , поскольку сеть $U_m^{2,(y,z),y'}$ должна решать задачу $I_{y'}^2(y,z)$.

Легко видеть, что

$$Q(U_m^{2,(y,z),y'}) = 2 \cdot |M_{y'}^2(y,z)| +]c \cdot |M_{y'}^2(y,z)|[\leq |M_{y'}^2(y,z)| \cdot (2 + c) + 1$$

и

$$T(U_m^{2,(y,z),y'}) < 2.$$

Объявим α' внутренней вершиной и уберем приписанную ей точку y' .

Теперь отождествим корень сети $U_m^{2,(y,z),y'}$ с вершиной α' , т.е. сеть $U_m^{2,(y,z),y'}$ теперь будет расти из α' , причем α' не будет полюсом полученной сети.

И, наконец, для каждого листа α'' сети $U_m^{2,(y,z),y'}$ заменим приписанную листу α'' точку y'' на четверку (y, z, y', y'') .

Проделаем такую операцию для каждого листа сети $U_m^{2,(y,z)}$.

Полученную сеть обозначим $U_{y,z}^2$.

Теперь объявим α внутренней вершиной, уберем приписанную ей пару (y, z) и отождествим α с корнем сети $U_{y,z}^2$, т.е. сеть $U_{y,z}^2$ будет расти из α .

Проделаем такую операцию для каждого листа сети U^1 .

Полученную сеть обозначим U^2 .

Сеть U^2 имеет не более $(k(k+1)/2)^2$ листьев, которым соответствуют четверки из множества Y^2 , причем каждая четверка $\tilde{y}^2 = (y_1^1, y_2^1, y_1^2, y_2^2)$ воспринимается как обозначение множества $V^3(\tilde{y}^2)$, определяемого соотношением (2.21).

Легко видеть, что $T(U^2) \leq 8$ и

$$\begin{aligned} Q(U^2) &\leq Q(U^1) + (2+c) \cdot \\ &\quad \cdot \left(\sum_{(y_1^1, y_2^1) \in Z^1} \left(|M^2(y_1^1, y_2^1)| + \sum_{y \in M^2(y_1^1, y_2^1)} |M_y^2(y_1^1, y_2^1)| \right) \right) + \\ &\quad + \left(\sum_{(y_1^1, y_2^1) \in Z^1} (1 + |M^2(y_1^1, y_2^1)|) \right) \leq \\ &\leq (2+c) \left(k + \frac{k(k+1)}{2} \right) + k + 1 + \\ &\quad + (2+c) \left(k \frac{k(k+1)}{2} + \left(\frac{k(k+1)}{2} \right)^2 \right) + \\ &\quad + (k+1) \frac{k(k+1)}{2} \leq \\ &\leq (2+c) \left(k(k+1)(k+2) + \left(\frac{k(k+1)}{2} \right)^2 \right). \end{aligned}$$

Далее из U^2 аналогичным образом получим сеть U^3 и т.д.

На $(n-1)$ -м шаге мы получим сеть U^{n-1} , которая будет иметь не более $(k(k+1)/2)^{n-1}$ листьев, которым соответствуют вектора из множества Y^{n-1} . Эта сеть позволяет для любого запроса

$$\tilde{x} = (u_1, v_1, \dots, u_n, v_n)$$

находить вектор $\tilde{y}^{n-1} = (y_1^1, y_2^1, \dots, y_1^{n-1}, y_2^{n-1}) \in Y^{n-1}$ такой, что y_1^i — ближайшая справа к u_i точка из $M^i(\tilde{y}^{i-1})$, а y_2^i — ближайшая слева к v_i точка из $M_{y_1^i}^i(\tilde{y}^{i-1})$ ($i = \overline{1, n-1}$), если, конечно, такой вектор существует. Если же такой вектор не существует, то ответ на запрос будет пуст, т.е. процесс поиска прервется, не доходя листьев. Если напомнить, что каждый вектор \tilde{y}^{i-n} воспринимается как обозначение множества $V^n(\tilde{y}^{n-1})$, то сеть U^{n-1} позволяет находить для любого запроса \tilde{x} подмножество точек множества V , которые удовлетворяют первым $n-1$ парам неравенств из (2.15).

Легко видеть, что

$$T(U^{n-1}) \leq 4(n-1)$$

и

$$Q(U^{n-1}) \leq Q(U^{n-2}) +$$

$$\begin{aligned}
& + (2+c) \left(k \cdot \left(\frac{k(k+1)}{2} \right)^{n-2} + \left(\frac{k(k+1)}{2} \right)^{n-1} \right) + \\
& + \left(\frac{k(k+1)}{2} \right)^{n-2} (k+1) \leq \\
\leq & (2+c) \left((k+2) \cdot (k(k+1))^{n-2} + \left(\frac{k(k+1)}{2} \right)^{n-1} \right).
\end{aligned}$$

Теперь осталось сделать последний шаг.

Пусть \mathcal{A} — множество листьев сети U_{n-1} .

Рассмотрим произвольный лист $\alpha \in \mathcal{A}$.

Пусть ему приписан вектор $\tilde{y}_\alpha^{n-1} = (y_1^1, y_2^1, \dots, y_1^{n-1}, y_2^{n-1})$, который является обозначением множества $V^n(\tilde{y}_\alpha^{n-1})$. Построим для проекции этого множества на последнюю координату y_n сеть U_α , решающую одномерную задачу интервального поиска, которую обозначим I_α . Поскольку в вершину α попадают только запросы из множества $X_\alpha = X^n(\tilde{y}_\alpha^{n-1})$, определяемого соотношением (2.44), то функция плотности вероятности для задачи I_α будет иметь вид $p_{\tilde{y}_\alpha^{n-1}}^n(u_n, v_n)$, описанный в соотношении (2.32), и эта функция в силу специальной ограниченности не больше, чем c . Сеть U_α построим по методу, описанному в предыдущем пункте. Тогда согласно доказанному

$$Q(U_\alpha) < 4 |V^n(\tilde{y}_\alpha^{n-1})| - 1 + 6 \cdot [\log(|V^n(\tilde{y}_\alpha^{n-1})|)] \cdot c \leq 4k - 1 + 6c \cdot [\log k].$$

Отметим также, что $T(U_\alpha) \leq R(I_\alpha) + 5$, где

$$R(I_\alpha) = \sum_{y \in M^n(\tilde{y}_\alpha^{n-1})} \mathbf{P}_\alpha(O(y, \rho_{int})).$$

Здесь \mathbf{P}_α — вероятностная мера на X_1 , задаваемая функцией плотности вероятности $p_{\tilde{y}_\alpha^{n-1}}^n(u_n, v_n)$.

Так как для любого $y \in M^n(\tilde{y}_\alpha^{n-1})$ величина $\mathbf{P}_\alpha(O(y, \rho_{int}))$ равна условной вероятности события $O(\tilde{y}, \rho_{intn})$ при условии события X_α , где \tilde{y} — такая запись из $V^n(\tilde{y}_\alpha^{n-1})$, что ее последняя координата равна y , то

$$\begin{aligned}
R(I_\alpha) &= \sum_{\tilde{y} \in V^n(\tilde{y}_\alpha^{n-1})} \mathbf{P}(O(\tilde{y}, \rho_{intn}) \cap X_\alpha) / \mathbf{P}(X_\alpha) = \\
&= \frac{1}{\mathbf{P}(X_\alpha)} \sum_{\tilde{y} \in V} \mathbf{P}(O(\tilde{y}, \rho_{intn}) \cap X_\alpha). \tag{2.46}
\end{aligned}$$

Возможность расширения области суммирования связана с тем, что для любой записи $\tilde{y} \in V \setminus V^n(\tilde{y}_\alpha^{n-1})$ вероятность

$$\mathbf{P}(O(\tilde{y}, \rho_{intn}) \cap X_\alpha) = 0.$$

Объявим теперь лист α внутренней вершиной, уберем приписанный ему вектор и прикрепим к нему сеть U_α .

Проделаем эту операцию для каждого листа сети U_{n-1} и полученную сеть обозначим U_n .

Так как в сети U_{n-1} не более $(k(k+1)/2)^{n-1}$ листьев, то

$$Q(U_n) \leq (4k + 2 + (1 + 6[\log k]) \cdot c) (k(k+1)/2)^{n-1}.$$

Легко видеть, что полученная сеть U_n разрешает задачу I , так как для любого запроса $\tilde{x} = (u_1, v_1, \dots, u_n, v_n)$ с помощью сети U^{n-1} мы находим подмножество точек множества V , которые удовлетворяют первым $n - 1$ парам неравенств из (2.15), а с помощью сети, растущей из вершины подсети U^{n-1} , соответствующей этому подмножеству, мы выберем из этого подмножества точки, удовлетворяющие и последней паре неравенств из (2.15).

Для того чтобы подсчитать сложность сети U^n , заметим, что для любых $\alpha, \alpha' \in \mathcal{A}$ таких, что $\alpha \neq \alpha'$, справедливо

$$X_\alpha \cap X_{\alpha'} = \emptyset$$

и

$$\bigcap_{\alpha \in \mathcal{A}} X_\alpha = X_{intn}.$$

Заметим также, что для любого запроса \tilde{x} существует только один активный путь, ведущий к концевым вершинам подсети U^{n-1} . Это означает, что для любого запроса \tilde{x} будет активна только одна из подсетей U_α , и, значит,

$$\begin{aligned} T(U^n) &\leq 4(n-1) + \sum_{\alpha \in \mathcal{A}} P(X_\alpha) \cdot (5 + R(I_\alpha)) = \\ &= 4n + 1 + \sum_{\alpha \in \mathcal{A}} P(X_\alpha) \cdot R(I_\alpha). \end{aligned}$$

Покажем, что

$$\sum_{\alpha \in \mathcal{A}} P(X_\alpha) \cdot R(I_\alpha) = R(I).$$

В самом деле, согласно (2.46)

$$\begin{aligned} \sum_{\alpha \in \mathcal{A}} P(X_\alpha) \cdot R(I_\alpha) &= \sum_{\alpha \in \mathcal{A}} \sum_{\tilde{y} \in V} \mathbf{P}(O(\tilde{y}, \rho_{intn}) \cap X_\alpha) = \\ &= \sum_{\tilde{y} \in V} \sum_{\alpha \in \mathcal{A}} \mathbf{P}(O(\tilde{y}, \rho_{intn}) \cap X_\alpha) = \\ &= \sum_{\tilde{y} \in V} \mathbf{P}\left(\bigcup_{\alpha \in \mathcal{A}} (O(\tilde{y}, \rho_{intn}) \cap X_\alpha)\right) = \\ &= \sum_{\tilde{y} \in V} \mathbf{P}(O(\tilde{y}, \rho_{intn})) = R(I). \end{aligned}$$

И, наконец, поскольку согласно теореме 4

$$T(I, \mathcal{F}) \geq R(I),$$

то теорема полностью доказана.

2.4.3 Пример оценки константы специальной ограниченности

Введенное в данном разделе свойство специальной ограниченности формулируется несколько громоздко и использовалось лишь с целью облегчения доказательства теоремы. Но практически все неэкзотические распределения обладают этим свойством с той или иной константой, и вопрос только в том, чтобы эту константу подсчитать.

В качестве примера подсчитаем константу специальной ограниченности для равномерного распределения вероятности.

Лемма 5 *Если функция плотности вероятности $p(\tilde{x})$ задает равномерную вероятностную меру на X_{intn} , то $p(\tilde{x})$ обладает свойством специальной ограниченности с константой 2 для любой конечной библиотеки, содержащейся в Y_{intn} .*

Доказательство. Так как $p(\tilde{x})$ задает равномерную вероятностную меру на X_{intn} , то

$$p(\tilde{x}) \equiv c = const.$$

Так как $p(\tilde{x})$ задает вероятностную меру на X_{intn} , то

$$\int_{X_{intn}} p(\tilde{x}) d\tilde{x} = 1.$$

Откуда

$$\begin{aligned} c \int_{X_{intn}} d\tilde{x} &= 1; \\ c \int_0^1 du_1 \int_{u_1}^1 dv_1 \cdots \int_0^1 du_n \int_{u_n}^1 dv_n &= 1; \\ c/2^n &= 1; \\ c &= 2^n. \end{aligned}$$

Легко видеть, что если $p_i(u_1, v_1, \dots, u_i, v_i)$ — функция, определяемая соотношением (2.29), где $i \in \{\overline{1, n-1}\}$, то для любого $(u_1, v_1, \dots, u_i, v_i) \in \underbrace{X_1 \times \cdots \times X_1}_i$ справедливо

$$p_i(u_1, v_1, \dots, u_i, v_i) = \int_0^1 du_n \int_{u_n}^1 dv_n \cdots \int_0^1 du_{i+1} \int_{u_{i+1}}^1 p(\tilde{x}) dv_{i+1} = 2^i. \quad (2.47)$$

Возьмем произвольную конечную библиотеку $V \supseteq Y_{intn}$.

Пусть Y^i — множества, определяемые соотношением (2.18) для этой библиотеки, $\tilde{y}^{i-1} = (y_1^1, y_2^1, \dots, y_1^{i-1}, y_2^{i-1}) \in Y^{i-1}$ и $p_{\tilde{y}^{i-1}}^i(u_i, v_i)$ — функции, определяемые соотношениями (2.31), где $i \in \{\overline{1, n}\}$. Тогда легко видеть, что для любого $i \in \{\overline{2, n}\}$ и любого $(u_i, v_i) \in X_1$ справедливо

$$\begin{aligned} p_{\tilde{y}^{i-1}}^i(u_i, v_i) &= \left. \int_{X'_{y_1^{i-1}, y_2^{i-1}}(M^{i-1}(\tilde{y}^{i-2}))}^{X'_{y_1^{i-1}, y_2^{i-1}}(M^{i-1}(\tilde{y}^{i-2}))} d\tilde{z}_{i-1} \cdots \int_{X'_{y_1^1, y_2^1}(M^1)}^{X'_{y_1^1, y_2^1}(M^1)} 2^i d\tilde{z}_1 \right/ \\ &\quad \left/ \int_{X'_{y_1^{i-1}, y_2^{i-1}}(M^{i-1}(\tilde{y}^{i-2}))}^{X'_{y_1^{i-1}, y_2^{i-1}}(M^{i-1}(\tilde{y}^{i-2}))} d\tilde{z}_{i-1} \cdots \right. \\ &\quad \left. \cdots \int_{X'_{y_1^1, y_2^1}(M^1)}^{X'_{y_1^1, y_2^1}(M^1)} 2^{i-1} d\tilde{z}_1 \right. = 2; \end{aligned} \quad (2.48)$$

$$p^1(u_1, v_1) = p_1(u_1, v_1) = 2 \quad (2.49)$$

в силу (2.47).

Откуда для функции $p_{\tilde{y}^{i-1}}^{i,1}(u_i)$, определяемой соотношением (2.33), для любого $i \in \{\overline{1, n-1}\}$ и любого $u_i \in [0, 1]$ справедливо

$$p_{\tilde{y}^{i-1}}^{i,1}(u_i) = \int_{u_i}^1 2dv_i = 2(1 - u_i) \leq 2. \quad (2.50)$$

Если обозначить через b_i начало, а через e_i конец отрезка $Z_{y_i}^1(M^i(\tilde{y}^{i-1}))$, то для функции $p_{\tilde{y}^{i-1}, y_i}^{i,2}(v_i)$, определяемой соотношением (2.34), для любого $i \in \{\overline{1, n-1}\}$, любого $v_i \in [b_i, 1]$

и любого $y_i \in M^i(\tilde{y}^{i-1})$ справедливо

$$\begin{aligned}
 p_{\tilde{y}^{i-1}, y_i}^{i, 2}(v_i) &= \int_{b_i}^{e_i} 2du_i / \int_{b_i}^{e_i} du_i \int_{u_i}^1 2dv_i = \\
 &= \int_{b_i}^{e_i} du_i / \int_{b_i}^{e_i} (1 - u_i) du_i = \\
 &= (e_i - b_i) / ((e_i - b_i) - (e_i^2/2 - b_i^2/2)) = \\
 &= 2/(2 - e_i - b_i) \leq 2/(1 - b_i).
 \end{aligned} \tag{2.51}$$

Осталось заметить, что соотношения (2.48) – (2.51) полностью доказывают утверждение леммы.

Следствие 1 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, определяемое соотношениями (2.35)–(2.42), $n \geq 1$, \mathbf{P} является равномерной вероятностной мерой на X_{intn} , и

$$R(I) = \sum_{y \in V} \mathbf{P}(O(y, \rho_{intn})).$$

Тогда

$$\begin{aligned}
 R(I) &< T(I, \mathcal{F}, (4k + 4 + 12[\log k])(k(k+1)/2)^{n-1}) \leq \\
 &\leq R(I) + 4n + 1.
 \end{aligned}$$

Данное утверждение является следствием леммы 5 и теоремы 8.

Отметим, что утверждения для многомерного случая задачи интервального поиска, приведенные в [16, 18], отличаются от утверждения теоремы 8 в общем случае, но совпадают в случае равномерного распределения. Отметим также, что доказательства теорем для многомерного случая задачи интервального поиска, приведенные в [16, 18], с целью сокращения объема отражают только идею доказательства. Полное доказательство этих теорем можно вести по методу, описанному в следующем разделе для многомерной задачи о доминировании. Но в связи с более сложной геометрией множества запросов в задаче интервального поиска доказательство будет очень громоздким. Поэтому мы не приводим здесь это доказательство, тем более, что по порядку в обоих случаях оценки объемов совпадают.

2.5 Задача о доминировании

2.5.1 Основные понятия и формулировка результата

Многомерная 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, где $n \geq 1$. 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:

$$(x_1, x_2, \dots, x_n) \rho_{dom} (y_1, y_2, \dots, y_n) \iff y_i \leq x_i, i = \overline{1, n}. \tag{2.52}$$

Тогда тип

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

назовем типом задачи о доминировании.

Пусть

$$\begin{aligned} G_1 &= \{ g_{i,m}^1(x_1, \dots, x_n) = \\ &= \max(1,]x_i \cdot m[) : i \in \{\overline{1, n}\}, m \in \mathbf{N} \}, \end{aligned} \quad (2.53)$$

$$\begin{aligned} G_2 &= \{ g_{i,a}^2(x_1, \dots, x_n) = \\ &= \begin{cases} 1, & \text{если } x_i \leq a \\ 2, & \text{если } x_i > a \end{cases} : i \in \{\overline{1, n}\}, a \in [0, 1] \}. \end{aligned} \quad (2.54)$$

Обозначим

$$M_a = \{x = (x_1, \dots, x_n) \in X_{dom} : x_n \geq a\}.$$

Пусть

$$F_1 = \{K_a : N_{K_a} = M_a, 0 \leq a \leq 1\}. \quad (2.55)$$

Пусть

$$\mathcal{F} = \langle F_1, G_1 \cup G_2 \rangle. \quad (2.56)$$

Справедлива следующая теорема.

Теорема 9 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, определяемое соотношениями (2.53)–(2.56), $n \geq 1$. Пусть

$$R(I) = \sum_{y \in V} \mathbf{P}(O(y, \rho_{dom})).$$

Тогда, если функция плотности вероятности $p(x) \leq c$, то

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

2.5.2 Мгновенное решение задачи о доминировании

В данном разделе мы приведем алгоритм, обеспечивающий мгновенное решение задачи о доминировании, и тем самым докажем теорему 9, поскольку нижняя оценка следует из теоремы 4.

Так же как и в случае с задачей интервального поиска мы рассмотрим отдельно две возможности: одномерную задачу о доминировании и многомерную.

Одномерный случай

В данном пункте мы рассмотрим одномерную задачу о доминировании, то есть случай, когда $n = 1$.

Пусть $V = \{y^1, \dots, y^k\}$, где $y^i \in [0, 1]$ $i = \overline{1, k}$. Будем считать, что записи в V упорядочены в порядке возрастания, то есть

$$y^1 \leq y^2 \leq \dots \leq y^k.$$

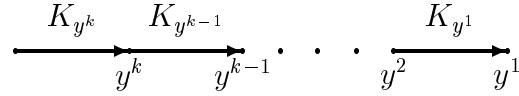


Рис. 2.2: Сеть для одномерной задачи о доминировании

Остается заметить, что отношение ρ_{dom} при $n = 1$ есть отношение линейного квазипорядка. Поэтому, как показано в доказательстве теоремы 5, ИСП U , изображенная на рисунке 2.2, разрешает ЗИП I , и ее сложность равна

$$T(U) = 1 + \sum_{i=1}^{k-1} \mathbf{P}(O(y^i, \rho_{dom})) \leq R(I) + 1,$$

а объем равен $Q(U) = n$.

Многомерный случай

В данном пункте мы рассмотрим многомерную задачу о доминировании, то есть случай, когда $n > 1$.

Введем вспомогательные обозначения.

Если $V' \subseteq Y_{dom}$, $r \in [0, 1]$ и $1 \leq i_1 < \dots < i_l \leq n$, то обозначим

$$S_r^i(V') = \{y = (y_1, y_2, \dots, y_n) \in V' : y_i \leq r\},$$

$$\Pi^{i_1, \dots, i_l}(V') = \{(y_{i_1}, \dots, y_{i_l}) : (y_1, \dots, y_n) \in V'\}.$$

Пусть $\tilde{y}^i = (y^1, y^2, \dots, y^i)$, где $i = \overline{0, n-1}$. Здесь и далее под значком \tilde{y}^0 будем понимать пустое место или отсутствие значка; так, например, запись $M^1(\tilde{y}^0)$ будем понимать как M^1 .

Введем по индукции следующие обозначения.

Базис индукции. Обозначим

$$V^1 = V, \quad M^1 = \Pi^1(V^1), \quad Y^1 = M^1, \quad X^1 = X_{dom}.$$

Шаг индукции. Пусть $i \in \{\overline{2, n}\}$ и для всех $j \in \{\overline{1, i-1}\}$ определены множества

$$V^j(\tilde{y}^{j-1}), \quad M^j(\tilde{y}^{j-1}), \quad Y^j, \quad X^j(\tilde{y}^{j-1}),$$

где $\tilde{y}^{j-1} = (y^1, y^2, \dots, y^{j-1})$ — произвольный вектор из Y^{j-1} . Пусть $\tilde{y}^{i-1} = (y^1, y^2, \dots, y^{i-1})$ — произвольный вектор из Y^{i-1} . Тогда обозначим

$$V^i(\tilde{y}^{i-1}) = S_{y^{i-1}}^{i-1}(V^{i-1}(y^1, \dots, y^{i-2})),$$

$$M^i(\tilde{y}^{i-1}) = \Pi^i(V^i(\tilde{y}^{i-1})),$$

$$Y^i = \{(y^1, \dots, y^i) : (y^1, \dots, y^{i-1}) \in Y^{i-1} \text{ и } y^i \in M^i(y^1, \dots, y^{i-1})\},$$

$$X^i(\tilde{y}^{i-1}) = \{ (x^1, \dots, x^n) \in X^{i-1}(y^1, \dots, y^{i-2}) : \\ x_{i-1} \in Z_{y^{i-1}}^2(M^{i-1}(y^1, \dots, y^{i-2})) \},$$

где $Z_y^2(M)$ определяется соотношением (2.25).

Построим по индукции некоторую информационную сеть U_q ($q \in \{\overline{1, n-1}\}$), которая удовлетворяет следующим условиям:

1. Сеть U_q имеет $|Y^q|$ листьев.
2. Между Y^q и множеством листьев сети U_q можно установить взаимно однозначное соответствие, такое, что если листу α сети U_q соответствует вектор (y^1, \dots, y^q) из Y^q , то
$$N_{\varphi_\alpha} = X^{q+1}(y^1, \dots, y^q).$$
3. $T(U_q) \leq 2q.$
4. $Q(U_q) \leq (2 + c) \cdot \sum_{i=1}^q |Y^i| + \sum_{i=1}^{q-1} |Y^i| + 1.$

Базис индукции. $q = 1$.

Возьмем множество M^1 и упорядочим его по возрастанию. Построим для него сеть U' , решающую вторую задачу о близости, так, как было описано в доказательстве теоремы 7, где в качестве параметра m возьмем $m = \lceil c \cdot |M^1| \rceil$.

Заменим в сети U' переключатель g_m^1 на $g_{1,m}^1$, а все переключатели вида $g_{y_\beta}^2$ на g_{1,y_β}^2 , где переключатели $g_{1,m}^1$ и g_{1,y_β}^2 описываются соотношениями (2.53) и (2.54) соответственно. Полученную сеть обозначим U_1 . Покажем, что сеть U^1 удовлетворяет условиям утверждения индукции.

Так как $Y^1 = M^1$, то сеть U_1 имеет $|Y^1|$ листьев и, следовательно, удовлетворяет условию 1 утверждения индукции.

Легко видеть, что в результате смены нагрузки переключательных вершин сеть U_1 на множество запросов X^1 решает следующую задачу о близости: она позволяет для произвольного запроса $x = (x_1, \dots, x_n) \in X^1$ находить такую точку из M^1 , которая является ближайшей слева к числу x_1 , то есть к первой координате вектора x . Откуда сразу следует, согласно теореме 1, что если y — произвольная точка из M^1 (а значит, и из Y^1) и α — лист, которому соответствует точка y , то

$$N_{\varphi_\alpha} = X^2(y),$$

что и доказывает выполнение условия 2 утверждения индукции.

Подсчитаем сложность сети U_1 . Обозначим

$$\begin{aligned} D_1 &= [0, 1/m], \\ D_i &= \left(\frac{i-1}{m}, \frac{i}{m} \right], \quad i = \overline{2, m}. \end{aligned} \tag{2.57}$$

Пусть

$$X_i = \{(x_1, \dots, x_n) \in X^1 : x_1 \in D_i\}, \quad i = \overline{1, m}.$$

Тогда так же, как в доказательстве теоремы 7, можно получить, что

$$T(U_1) \leq 1 + \sum_{i=1}^m L_2(|V_i|) \cdot \mathbf{P}(X_i),$$

где $V_i = M^1 \cap D_i$. Так как

$$\mathbf{P}(X_i) = \int_{D_i} dx_1 \int_0^1 \cdots \int_0^1 p(x_1, x_2, \dots, x_n) dx_2 \cdots dx_n \leq c/m,$$

то согласно лемме 4

$$T(U_1) \leq 1 + \frac{c}{m} \sum_{i=1}^m L_2(|V_i|) \leq 1 + \frac{c|M^1|}{\lceil c|M^1| \rceil} < 2.$$

Что и доказывает выполнение условия 3 утверждения индукции.

Объем сети U_1 был подсчитан в доказательстве теоремы 7, и он равен

$$Q(U_1) = m + 2|M^1| \leq (2 + c) \cdot |Y^1| + 1.$$

Следовательно, выполняется условие 4 утверждения индукции.

Индуктивный переход. Пусть $1 < q < n$ и построена сеть U_{q-1} , удовлетворяющая условиям индукции. Покажем, как можно построить сеть U_q .

Возьмем произвольный лист α сети U_{q-1} . Пусть ему соответствует вектор (y^1, \dots, y^{q-1}) из Y^{q-1} . Упорядочим по возрастанию множество $M^q(y^1, \dots, y^{q-1})$. Построим для него сеть U_α , решающую вторую задачу о близости, так, как было описано в доказательстве теоремы 7, где в качестве параметра m возьмем $m = [c \cdot |M^q(y^1, \dots, y^{q-1})|]$.

Заменим в сети U_α переключатель g_m^1 на $g_{q,m}^1$, а все переключатели вида $g_{y_\beta}^2$ на g_{q,y_β}^2 , где переключатели $g_{q,m}^1$ и g_{q,y_β}^2 описываются соотношениями (2.53) и (2.54) соответственно.

Уберем нагрузку листа α и объявим его обычной вершиной, после чего отождествим его с корнем сети U_α , то есть U_α будет расти из α .

Проделаем такую операцию для каждого листа сети U_{q-1} и полученную в результате сеть обозначим U_q . Покажем, что сеть U^q удовлетворяет условиям утверждения индукции.

Поскольку множество Y^q строится таким образом, что каждый вектор (y^1, \dots, y^{q-1}) из Y^{q-1} порождает $|M^q(y^1, \dots, y^{q-1})|$ векторов множества Y^q , и поскольку из каждого листа сети U_{q-1} , которому соответствует вектор (y^1, \dots, y^{q-1}) из Y^{q-1} , выпускается сеть, имеющая $|M^q(y^1, \dots, y^{q-1})|$ листьев, то сеть U_q имеет ровно $|Y^q|$ листьев; тем самым доказано выполнение условия 1 утверждения индукции.

Покажем, что выполняется условие 2 утверждения индукции. Для этого рассмотрим произвольный лист α сети U_{q-1} . Пусть ему соответствовал вектор (y^1, \dots, y^{q-1}) из Y^{q-1} . Согласно предположению индукции функция фильтра листа α такова, что пропускает в лист α только запросы из множества $X^q(y^1, \dots, y^{q-1})$. Легко видеть, что в результате смены нагрузки переключательных вершин сеть U_α решает следующую задачу о близости: она позволяет для произвольного запроса $x = (x_1, \dots, x_n) \in X^q(y^1, \dots, y^{q-1})$ (поскольку только эти запросы достигают листа α) находить такую точку из $M^q(y^1, \dots, y^{q-1})$, которая является ближайшей слева к числу x_q , то есть к q -й координате вектора x . Откуда сразу следует, согласно теореме 1, что если y — произвольная точка из $M^q(y^1, \dots, y^{q-1})$ (и значит (y^1, \dots, y^{q-1}, y) принадлежит Y^q) и α' — лист, которому соответствует точка y , то

$$N_{\varphi_{\alpha'}} = X^{q+1}(y^1, \dots, y^{q-1}, y)$$

и листу α' можно сопоставить вектор (y^1, \dots, y^{q-1}, y) из Y^q , что и доказывает выполнение условия 2 утверждения индукции.

Подсчитаем сложность сети U_q .

Возьмем произвольный лист α сети U_{q-1} . Пусть ему соответствовал вектор (y^1, \dots, y^{q-1}) из Y^{q-1} . Согласно предположению индукции, только запросы из множества $X^q(y^1, \dots, y^{q-1})$ достигают листа α . Переобозначим это множество в X^α . Переобозначим множество $M^q(y^1, \dots, y^{q-1})$ в M^α , а множества $Z_{y^i}^2(M^i(y^1, \dots, y^{i-1}))$ — в Z_i^α , где $i = \overline{1, q-1}$. Тогда легко видеть, что

$$X^\alpha = Z_1^\alpha \times \cdots \times Z_{q-1}^\alpha \times \underbrace{[0, 1] \times \cdots \times [0, 1]}_{n-q+1}.$$

Подсчитаем сложность сети U_α . Обозначим

$$X_i^\alpha = \{(x_1, \dots, x_n) \in X^\alpha : x_q \in D_i\}, \quad i = \overline{1, m},$$

где $m = \lceil c|M^\alpha| \rceil$, D_i определяются соотношениями (2.57). Тогда так же, как в доказательстве теоремы 7, с учетом того, что в α попадают только запросы из X^α , можно получить, что

$$T(U_\alpha) \leq P(X^\alpha) + \sum_{i=1}^m L_2(|V_i^\alpha|) \cdot \mathbf{P}(X_i^\alpha),$$

где $V_i^\alpha = M^\alpha \cap D_i$. Так как

$$\begin{aligned} \mathbf{P}(X_i^\alpha) &= \int_{Z_1^\alpha} dx_1 \cdots \int_{Z_{q-1}^\alpha} dx_{q-1} \int_{D_i} dx_q \int_0^1 \cdots \\ &\quad \cdots \int_0^1 p(x_1, \dots, x_n) dx_{q+1} \cdots dx_n \leq \frac{c}{m} \prod_{j=1}^{q-1} l(Z_j^\alpha), \end{aligned}$$

где через $l(Z_j^\alpha)$ обозначена длина отрезка Z_j^α , то согласно лемме 4

$$\begin{aligned} T(U_\alpha) &\leq P(X^\alpha) + \frac{c}{m} \prod_{j=1}^{q-1} l(Z_j^\alpha) \sum_{i=1}^m L_2(|V_i|) \leq \\ &\leq P(X^\alpha) + \frac{c|M^1|}{\lceil c|M^1| \rceil} \prod_{j=1}^{q-1} l(Z_j^\alpha) < P(X^\alpha) + \prod_{j=1}^{q-1} l(Z_j^\alpha). \end{aligned}$$

Теперь заметим, что если $\alpha, \alpha' \in \mathcal{L}(U_{q-1})$ (то есть α и α' — листья сети U_{q-1}) и $\alpha \neq \alpha'$, то $X^\alpha \cap X^{\alpha'} = \emptyset$. Кроме того,

$$\bigcup_{\alpha \in \mathcal{L}(U_{q-1})} X^\alpha \subseteq X_{dom}.$$

Откуда следует, что

$$\sum_{\alpha \in \mathcal{L}(U_{q-1})} \mathbf{P}(X^\alpha) \leq 1$$

и

$$\sum_{\alpha \in \mathcal{L}(U_{q-1})} \prod_{j=1}^{q-1} l(Z_j^\alpha) \leq 1. \tag{2.58}$$

Неравенство (2.58) следует из того, что величина $\prod_{j=1}^{q-1} l(Z_j^\alpha)$ равна объему многомерного параллелепипеда X^α , а сумма объемов всех параллелепипедов X^α не превышает объема параллелепипеда X_{dom} , то есть 1.

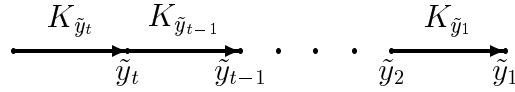
Следовательно,

$$\begin{aligned} T(U_q) &= T(U_{q-1}) + \sum_{\alpha \in \mathcal{L}(U_{q-1})} T(U_\alpha) \leq \\ &\leq T(U_{q-1}) + \sum_{\alpha \in \mathcal{L}(U_{q-1})} \left(P(X^\alpha) + \prod_{j=1}^{q-1} l(Z_j^\alpha) \right) \leq \\ &\leq 2(q-1) + 2 = 2q. \end{aligned}$$

Таким образом, условие 3 утверждения индукции выполняется.

Подсчитаем объем сети U_q .

$$\begin{aligned} Q(U_q) &= Q(U_{q-1}) + \sum_{\alpha \in \mathcal{L}(U_{q-1})} Q(U_\alpha) \leq \\ &\leq Q(U_{q-1}) + \sum_{\alpha \in \mathcal{L}(U_{q-1})} ((2+c)|M^\alpha| + 1) \leq \end{aligned}$$

Рис. 2.3: Сеть U'_α

$$\begin{aligned} &\leq (2+c) \cdot \sum_{i=1}^{q-1} |Y^i| + \sum_{i=1}^{q-2} |Y^i| + 1 + (2+c)|Y^q| + |Y^{q-1}| = \\ &= (2+c) \cdot \sum_{i=1}^q |Y^i| + \sum_{i=1}^{q-1} |Y^i| + 1. \end{aligned}$$

Следовательно, условие 4 утверждения индукции выполняется. И, значит, нам удалось построить сеть U_q , которая удовлетворяет всем условиям утверждения индукции.

Строить сеть U , обеспечивающую мгновенное решение многомерной задачи о доминировании I , будем следующим образом. Возьмем сеть U_{n-1} и рассмотрим произвольный лист α этой сети. Пусть ему соответствует вектор (y^1, \dots, y^{n-1}) из Y^{n-1} . Возьмем множество $V^n(y^1, \dots, y^{n-1})$ и переобозначим его в V^α . Пусть

$$V^\alpha = \{\tilde{y}_1, \dots, \tilde{y}_t\},$$

где записи упорядочены по последней координате. Построим сеть U'_α , изображенную на рисунке 2.3. Уберем нагрузку листа α и объявим его обычной вершиной, после чего отождествим его с корнем сети U'_α , то есть U'_α будет расти из α .

Проделаем такую операцию для каждого листа сети U_{n-1} и полученную в результате сеть обозначим U .

Покажем, что сеть U разрешает задачу I .

Пусть α — произвольный лист сети U_{n-1} и пусть ему соответствовал вектор (y^1, \dots, y^{n-1}) из Y^{n-1} . Переобозначим множество $X^n(y^1, \dots, y^{n-1})$ в X^α .

Возьмем произвольный запрос $x = (x_1, \dots, x_n) \in X_{dom}$. Рассмотрим два случая.

Первый случай: существует такой лист α сети U_{n-1} , что $x \in X^\alpha$. Это означает, что запрос x достигнет листа α и ни в какие другие листы сети U_{n-1} не попадет, так как для любых различных $\alpha', \alpha'' \in \mathcal{L}(U_{n-1})$ справедливо $X^{\alpha'} \cap X^{\alpha''} = \emptyset$. Из листа α исходит цепочка, вершинам которой сопоставлены записи из множества $V^n(y^1, \dots, y^{n-1})$, а это множество есть не что иное, как множество всех записей из V , у которых i -я координата не превышает y^i ($i = \overline{1, n-1}$). То, что $x \in X^\alpha$, означает, что y^1 — ближайшая слева точка к x_1 в M^1 , y^2 — ближайшая слева точка к x_2 в $M^2(y^1)$, которое представляет собой проекцию на вторую координату множества $V^2(y^1)$, и т.д. Таким образом, множество $V^n(y^1, \dots, y^{n-1})$ содержит все те и только те точки из V , которые удовлетворяют первым $n-1$ неравенствам (2.52). Далее запрос x продвигается вдоль цепочки, растущей из α , пока записи из $V^n(y^1, \dots, y^{n-1})$ не больше по последней координате, чем x_n . Тем самым ответ на запрос x будет содержать только те записи, которые удовлетворяют запросу, причем все такие записи.

Второй случай: для любого $\alpha \in \mathcal{L}(U_{n-1})$ запрос $x \notin X^\alpha$. Это означает, что запрос не дойдет ни до одного из листьев сети U_{n-1} , и, значит, ответ на запрос x будет пуст. Осталось показать, что при таком x в библиотеке V не содержится записей, удовлетворяющих x .

Пусть $i \in \{\overline{1, n-1}\}$ — такой номер, что существует такой набор (y^1, \dots, y^{i-1}) из Y^{i-1} , что $x \in X^i(y^1, \dots, y^{i-1})$, и для любого набора (y^1, \dots, y^i) из Y^i запрос $x \notin X^{i+1}(y^1, \dots, y^i)$. Очевидно, что такой номер есть, так как $x \in X^1 = X_{dom}$. Так как для некоторого набора (y^1, \dots, y^{i-1}) из Y^{i-1} выполняется $x \in X^i(y^1, \dots, y^{i-1})$ (очевидно, что такой набор только один), то запрос x пройдет в лист γ сети U_{i-1} , которому соответствует набор (y^1, \dots, y^{i-1}) .

(если $i = 1$, то в качестве γ рассматривается корень сети U). Из листа γ растет сеть, которая решает вторую задачу о близости для множества $M^i(y^1, \dots, y^{i-1})$, и так как для любого набора (y^1, \dots, y^i) из Y^i запрос $x \notin X^{i+1}(y^1, \dots, y^i)$, то это значит, что x_i меньше, чем любая точка из $M^i(y^1, \dots, y^{i-1})$, то есть в множестве $V^i(y^1, \dots, y^{i-1})$, которое состоит из записей, удовлетворяющих первым $i - 1$ неравенствам из (2.52), не найдется ни одной точки, i -я координата которой не больше, чем x_i . Следовательно, в V нет записей, удовлетворяющих запросу x .

Тем самым мы показали, что сеть U разрешает задачу I .

Подсчитаем сложность сети U .

Рассмотрим произвольный лист α сети U_{n-1} . Пусть ему соответствовал вектор (y^1, \dots, y^{n-1}) из Y^{n-1} . Цепочку ребер, растущую из α , обозначим через U'_α . Переобозначим множество $V^n(y^1, \dots, y^{n-1})$ в V^α , а $X^n(y^1, \dots, y^{n-1})$ в X^α .

Легко видеть, что сложность сети U'_α равна

$$T(U'_\alpha) = P(X^\alpha) + \sum_{\tilde{y} \in V} P(X^\alpha \cap O(\tilde{y}, \rho_{dom})).$$

Откуда

$$\begin{aligned} T(U) &= T(U_{n-1}) + \sum_{\alpha \in \mathcal{L}(U_{n-1})} T(U'_\alpha) \leq \\ &\leq 2(n-1) + \sum_{\alpha \in \mathcal{L}(U_{n-1})} \left(P(X^\alpha) + \sum_{\tilde{y} \in V} P(X^\alpha \cap O(\tilde{y}, \rho_{dom})) \right) \leq \\ &\leq 2(n-1) + \sum_{\alpha \in \mathcal{L}(U_{n-1})} P(X^\alpha) + \\ &\quad + \sum_{\tilde{y} \in V} \sum_{\alpha \in \mathcal{L}(U_{n-1})} P(X^\alpha \cap O(\tilde{y}, \rho_{dom})) \leq \\ &\leq 2(n-1) + 1 + \sum_{\tilde{y} \in V} P(O(\tilde{y}, \rho_{dom})) = 2n - 1 + R(I). \end{aligned}$$

Подсчитаем объем сети U . Для этого оценим величины $|Y^q|$, где $q = \overline{1, n}$. Понятно, что если библиотека V такова, что ее проекция на любую ось координат имеет мощность k , то величины $|Y^q|$ будут достигать максимума, и в этом случае эти величины будут зависеть только от k и q . Покажем индукцией по q , что для такого рода библиотек $|Y^q| = \binom{k+q-1}{q}$.

Базис индукции. $q = 1$. Тогда

$$|Y^1| = |M^1| = k = \binom{k}{1}.$$

Индуктивный переход. Пусть для любого $i \in \{\overline{1, q-1}\}$ справедливо $|Y^i| = \binom{k+i-1}{i}$. Напомним, что по построению количество листьев в сети U_q равно $|Y^q|$. Рассмотрим сеть U_1 . Пусть $Y^1 = M^1 = \{y_1, \dots, y_k\}$, где $y_1 < y_2 < \dots < y_k$. Обозначим через α_i лист сети U_1 , которому соответствует точка y_i , а через U_i^q — подсеть сети U_q , растущую из листа α_i . Легко заметить, что граф сети U_i^q изоморфен графу сети U_{q-1} при $k = i$ и, значит, число листьев сети U_i^q по предположению индукции равно $\binom{i+q-2}{q-1}$. Откуда число листьев сети U_q равно

$$|Y^q| = \sum_{i=1}^k \binom{i+q-2}{q-1} = \binom{k+q-1}{q}.$$

Последнее равенство можно найти в качестве упражнения, например, в [4, стр. 262].

Тем самым мы показали, что для любой библиотеки V и любого $q \in \{\overline{1, n}\}$ величина

$$|Y^q| \leq \binom{k+q-1}{q}.$$

По построению сети U легко видеть, что объем сети U равен объему сети U_{n-1} плюс количество листьев в сети U , так как помимо ребер сети U_{n-1} в сети U есть только ребра, ведущие в листья, и на каждый лист приходится ровно одно ребро. По аналогии с тем, как мы определяли число листьев в сети U_q , легко показать, что число листьев сети U равно $|Y^n|$. Следовательно,

$$\begin{aligned} Q(U) &= Q(U_{n-1}) + |Y^n| = \\ &= |Y^n| + (2+c) \cdot \sum_{i=1}^{n-1} |Y^i| + \sum_{i=1}^{n-2} |Y^i| + 1 \leq \\ &\leq \binom{k+n-1}{n} + (3+c) \cdot \sum_{i=1}^{n-1} \binom{k+i-1}{i}. \end{aligned}$$

Тем самым теорема 9 полностью доказана.

Литература

- [1] Абчук В. А., Сузdalь В. Г. Поиск объектов. М.: Советское радио, 1977.
- [2] Альсведе Р., Вегенер И. Задачи поиска. М.: Мир, 1982.
- [3] Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.
- [4] Гаврилов Г. П., Сапоженко А. А. Задачи и упражнения по курсу дискретной математики. М.: Наука, 1992.
- [5] Гасанов Э. Э. Некоторые оценки сложности поиска информации // Физическое и математическое моделирование дискретных систем. Межвузовский сборник трудов №56. М.: Изд-во Моск. энерг. ин-та, 1985. – С. 43–47.
- [6] Гасанов Э. Э. Алгоритмы построения информационных деревьев. Препринт Р-5-188 ИЯФ АН УзССР. Ташкент, 1985.
- [7] Гасанов Э. Э. Оценки средней сложности поиска информации. Препринт Р-5-186 ИЯФ АН УзССР. Ташкент, 1985.
- [8] Гасанов Э. Э. О сложности информационного поиска: Дис. ... канд. физ.-мат. наук. – М., 1985.
- [9] Гасанов Э. Э. О некоторых оценках сложности поиска информации // Алгебра, логика и теория чисел: Сб. статей. – М.: Изд-во МГУ, 1986. – С. 37–39.
- [10] Гасанов Э. Э. О виде оптимальных информационных сетей для отношений линейного квазипорядка. Препринт Р-5-303 ИЯФ АН УзССР. Ташкент, 1987.
- [11] Гасанов Э. Э. Оптимальные информационные сети для отношений поиска, являющихся отношениями линейного квазипорядка // Конструкции в алгебре и логике: Сб. статей. – Тверь, Изд-во Тверского гос. ун-та, 1990. – С. 11–17.
- [12] Гасанов Э. Э. Об одной математической модели информационного поиска // Дискретная математика. – 1991. – Т. 3. Вып. 2. – С. 69–76.
- [13] Гасанов Э. Э. Математические модели и сложность информационного поиска // Proceedings of the graduate workshop in mathematics and its applications in social sciences. Ljubljana. – 1991. – Р. 37–53.
- [14] 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.

- [15] Гасанов Э. Э. О сложности поиска в базах данных // Искусственный интеллект: Межвузовский сборник трудов. – Саратов, Изд-во Саратовского университета, 1993. – С. 41–56.
- [16] Гасанов Э. Э. Некоторые задачи поиска, допускающие мгновенное в среднем решение // Фундаментальная и прикладная математика. – 1995. – Т. 1. Вып. 1. – С. 123–146.
- [17] Гасанов Э. Э. Об одномерной задаче интервального поиска // Дискретная математика. – 1995. – Т. 7. Вып. 2. – С. 40–60.
- [18] Гасанов Э. Э. Мгновенно решаемые задачи поиска // Дискретная математика. – 1996. – Т. 8. Вып. 3. – С. 119–134.
- [19] Дейт К. Введение в системы баз данных. М.: Наука, 1980.
- [20] Ершов А.П. Некоторые субъективные замечания к актуальным проблемам программирования // Перспективы системного и теоретического программирования: Труды Всесоюзного симпозиума, Новосибирск, 20–22 марта, 1978 г. – ВЦ СО АН СССР, Новосибирск, 1979. – С. 113–127.
- [21] Информационные системы общего назначения. М.: Статистика, 1975.
- [22] Ким Д. П. Методы поиска и преследования подвижных объектов. М.: Наука, 1989.
- [23] Кнут Д. Искусство программирования для ЭВМ: Т.3: Сортировка и поиск. М.: Мир, 1978.
- [24] Ли Д., Препарата Ф. Вычислительная геометрия. Обзор // Кибернетический сб. – 1987. Вып. 24. – С. 5–96.
- [25] Мартин Дж. Организация баз данных в вычислительных системах. М.: Мир, 1980.
- [26] Наумов А. Н., Вендров А. М., Иванов В. К. и др. Системы управления базами данных и знаний: Справ. изд. М.: Финансы и статистика, 1991.
- [27] Ньюмен У. М., Спруэлл Р. Ф. Основы интерактивной машинной графики. М.: Мир, 1976.
- [28] Решетников В. Н. Алгебраическая теория информационного поиска // Программирование. – 1979. – № 3. – С. 68–74.
- [29] Селтон Г. Автоматическая обработка, хранение и поиск информации. М.: Советское радио, 1973.
- [30] Системы управления базами данных для ЕС ЭВМ: Справочник / Под общ. ред. В. М. Савинкова. М.: Финансы и статистика, 1984.
- [31] Солтон Дж. Динамические библиотечно-информационные системы. М.: Мир, 1979.
- [32] Тиори Т., Фрай Дж. Проектирование структур баз данных: В 2-х кн. М.: Мир, 1985.
- [33] Хеллман О. Введение в теорию оптимального поиска. М.: Наука, 1985.
- [34] Ben-Or M. Lower bounds for algebraic computation trees // Proc. 15th ACM Annu. Symp. Theory Comput. – Apr. 1983. – P. 80–86.

- [35] Chazelle B. M. Filtering search: a new approach to query-answering // Proc. 24th IEEE Annu. Symp. Found. Comput. Sci. – Nov. 1983. – P. 122–132.
- [36] CODASYL Systems Committee, Feature Analysis of Generalized Data Base Management Systems, ASM, New York, 1971b.
- [37] CODASYL Systems Committee, Selection and Acquisition Of Data Base Management Systems, ASM, New York, Mar. 1976.
- [38] CODASYL — The Stored-Data Definition and Translation Task Group, Stored-Data Description and Data Translation: A Model and Language // Inf. Syst. – 1977. – V. 2, №3. – P. 95–148.
- [39] Codd E. F. A Relation Model of Data for Large Shared Data Banks // Comm. ACM 13, №6, ACM, New York, London, Amsterdam, June 1970. P. 377–387.
- [40] Codd E. F. Further Normalization of the Data Base Relational Model // Courant Computer Sci. Symposia (vol. 6: "Data-Base System"), ed. by R. Rustin, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1972.
- [41] Codd E. F. Relational Completeness of Data Base Sublanguages // Courant Computer Sci. Symposia (vol. 6: "Data-Base System"), ed. by R. Rustin, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1972.
- [42] Codd E. F. A Data Base Sublanguage Founded on the Relational Calculus // Proc. of the 1971 ACM-SIGFIDET Workshop on Data Description, Access, and Control, ACM, New York, London, Amsterdam, 1972.
- [43] Codd E. F. Access Control for Relational Data Base Systems // BCS Symposium on Relational Data-Base Concepts, Apr. 1973, British Computer Soc., London, 1973.
- [44] Codd E. F. Recent Investigations in Relational Data-Base Systems // Information Processing'74, North-Holland, Amsterdam, 1974.
- [45] Codd E. F. Relational Database: A Practical Foundation for Productivity // Commun. of ACM. – 1982. – V. 25, №2. – P. 140–155.
- [46] Data Language/I-System/370 DOS/VS, General Information Manual GH20-1246, IBM, White Plains, New York, 1974.
- [47] Date C. J., Codd E. F. The Relational and Network Approaches: Comparison of the Application Programming Interfaces // Proc. of the 1974 ACM-SIGFIDET Workshop, ACM, New York, London, Amsterdam, 1974.
- [48] Dobkin D. P. A nonlinear lower bound on search tree programs for solving knapsack problems // J. Comput. Syst. Sci. – 1976. – V. 13. – P. 69–73.
- [49] Dobkin D. P., Lipton R. J. A lower bound of $1/2n^2$ on linear search programs for the knapsack problem // J. Comput. Sci. – 1978. – V. 16. – P. 413–417.
- [50] Dobkin D. P., Lipton R. J. On the complexity of computations under varying sets of primitives // J. Comput. Syst. Sci. – 1979. – V. 18. – P. 86–91.

- [51] Edelsbrunner H., Overmars M. H., Siedel R. Some methods of computational geometry applied to computer graphics // IIG, Technische Univ. Graz, Austria, Tech. Rep. F117. – June 1983.
- [52] Gasanov E. E. On fast solving of interval search problem // Proceedings of International Congress of Mathematicians. Zurich, Switzerland, 1994. – P. 137.
- [53] Gasanov E. E. Instantly solvable search problems // Proceedings of International Symposium on Intelligent Data Analysis (IDA-95). Baden-Baden, Germany. – IIAS Press, 1995. – P. 65–69.
- [54] Information Management System Virtual Storage (IMS/VS), General Information Manual GH20-1260, IBM, White Plains, New York, 1974.
- [55] Information Management System/360, Version 2, Application Programming Reference Manual SH20-0912, IBM, White Plains, New York, 1974.
- [56] Information Management System/360, Version 2, System Programming Reference Manual SH20-0911, IBM, White Plains, New York, 1974.
- [57] Lee D. T., Wong C. K. Quintari trees: A file structures for multidimensional database system // ACM Trans. Database Syst. – Sept. 1980. – V. 1, №1. – P. 339–353.
- [58] Steele J. M., Yao A. C. Lower bounds for algebraic decision trees // J. Algorith. – 1982. – V. 3. – P. 1–8.
- [59] Yao A. C., Rivest R. L. On the polyhedral decision problem // SIAM J. Comput. – 1980. – V. 9. – P. 343–347.