ОБЪЕДИНЕННЫЙ
ИНСТИТУТ
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

P11 - 9777

М.Д.Иванчев, В.М.Сумароков, Н.И.Янев

КОМБИНАТОРНЫЙ АНАЛИЗ СБАЛАНСИРОВАННЫХ ПРИЗНАКОВЫХ ДЕРЕВЬЕВ С ПЕРЕМЕННЫМ РАЗМЕРОМ ПУЧКА

1976

Р11 - 9777

М.Д.Иванчев, В.М.Сумароков, Н.И.Янев

КОМБИНАТОРНЫЙ АНАЛИЗ СБАЛАНСИРОВАННЫХ ПРИЗНАКОВЫХ ДЕРЕВЬЕВ С ПЕРЕМЕННЫМ РАЗМЕРОМ ПУЧКА

Направлено в журнал "Программирование"

В большинстве задач, связанных с машинной обработкой больших информационных массивов, часто возникает потребность в эффективной организации доступа к элементам массива по признаку, или ключу. В настоящее время развитие методов, обеспечивающих такой доступ, идет по двум основным направлениям: во-первых, по пути создания вычислительных схем, реализующих различные алгоритмы определения адресов машинных образов объектов по их признаку, и, во-вторых, на основе разработки структур, существенно использующих отношение лексикографического упорядочения между элементами массива.

Эволюция методов первой группы осуществляется в процессе преодоления двух характерных трудностей: I) уменьшения полисемии, вследствие которой для объектов с разными значениями признака функция расстановки назначает одно и то же поле памяти, и 2) сокращения областей неполного заполнения памяти в результате стохастичности расстановки. Для структурных методов характерны значительные затраты памяти на размещение управляющей "признаковой" информации и многочисленных адресов связи, посредством которых реализуются цепные списки. Поэтому в условиях ограниченных ресурсов памяти серийных ЭВМ возникает задача сокращения таких потерь.

Вместе с тем структура данных должна обладать развитыми динамическими свойствами, обеспечивающими, наряду с быстрым поиском, эффективные процедуры включения новой и исключения устаревшей информации. В этом смысле сложившийся интерес представляют наращиваемые признаковые деревья. Наиболее изученной их версией являются двусвязные дихотомические структуры, хорошо представленные в литературе / I, 2 /. Каждый узел в таких деревьях соответствует одному объекту и интерпретируется на четырех полях

памяти: значение поискового признака объекта, адрес информации об объекте (или некотором его описании) и два адреса связи, обеспечивающие ветвление дерева по мере увеличения массива.

В данной статье рассматриваются поисковые свойства наращиваемых признаковых деревьев, использующих для организации ветвления только один адрес связи. Это приводит к небольшому относительному замедлению поиска, а также к существенному сокращению затрат памяти – не только вследствие отказа от одного адреса связи, но и в силу следующего обстоятельства. В большинстве широко распространенных ЭВМ ограниченная разрядность машинного слова не позволяет разместить все четыре поля в одной ячейке при адресации объектов в пределах оперативной памяти. Однако во многих случаях она оказывается достаточной для размещения трех полей, поэтому реальная экономия памяти при использовании односвязных деревьев, без существенной потери быстродействия их программного обеспечения, – ставится двукратной.

Вопросы исследования и эффективной машинной реализации односвязных признаковых деревьев рассматриваются, например, в /1,3,4/. Ниже на основе методов, используемых в /5/, предпринято дальнейшее развитие вопроса.

I. Основные понятия

Дерево – это ориентированный граф без циклов, удовлетворяющий требованию связности: для любой пары узлов a_i и a_j существует путь $\rho(a_i, a_j)$ как непрерывная последовательность ребер, связывающих a_i и a_j. В каждый узел дерева, кроме корня, входит одно ребро и из каждого узла, кроме терминальных, выходит несколько ребер. Узлы, расположенные в конце путей одинаковой длины, принадлежат одному уровню. Номер уровня, которому принадлежит узел, определяется количеством ребер, соединяющих этот узел с корнем. Будем считать, что корень дерева находится на нулевом, а терминальные узлы – на m-ом уровне. Совокупность выходящих из узла ребер образует пучок, а их количество называется размером пучка.

Наращиваемое дерево строится для одного поискового признака, который имеет количественное выражение и значения которого являются идентификаторами узлов дерева. Поиск объекта по признаку в

информационном массиве, построенном в виде дерева, происходит в процессе сравнения со значениями признака, идентифицирующими узлы дерева, при спуске от корня к терминальному уровню. Траектория, описываемая при этом поисковой точкой, называется поисковым треком и зависит как от структуры дерева, так и от определенного на нем алгоритма поиска. Узел, которому соответствует объект или некоторое его описание, называется объектным; узел, идентифицируемый значением признака, но не содержащий объектной информации, называется признаковым. Каждый узел дерева всегда является признаковым, но не всегда – объектным. Последнее отражает широкое разнообразие существующих структурных схем признаковых деревьев (см., например, /1,3/).

2. Постановка задачи

Любой реальный объект представляется в информационной системе некоторым своим образом, характеристикой которого служит информация, выражаемая конкретным значением поискового признака. Всякая рациональная организация структуры наращиваемых деревьев предполагает размещение объектной информации таким образом, чтобы между соответствующими значениями поискового признака выполнялось отношение лексикографического порядка. Это позволяет сделать поиск целенаправленным за счет исключения из числа просматриваемых значительного множества узлов, идентификаторы (ключи) которых не соответствуют поисковому предписанию в отношении обусловленного порядка.

В односвязных наращиваемых деревьях Прайса, Грэя и Ландауэра /4/, описанных также в /3/, множество возможных значений признака не ограничивается и определяется фактическими значениями идентификаторов, поступающих в систему объектов. Пучок в корне дерева делит весь диапазон изменения поискового признака на несколько интервалов, каждый пучок следующего уровня дробит эти интервалы на еще более мелкие и т.д. Ключи последовательных узлов пучка образуют возрастающую последовательность значений признака, поэтому отношение порядка сохраняется для всех узлов на каждом из уровней.

Если при поступлении объектов в систему имеются тенденции монотонного представления соответствующих поисковых ключей, то ветви дерева начинают расти преимущественно в некоторых направ-

лениях. Это увеличивает трудоемкость доступа к данным. В /4/ описан алгоритм балансировки односвязного дерева, начинающий формирование нового уровня только по заполнении предыдущего, что позволяет сократить трудоемкость поиска в среднем. Мы будем также рассматривать сбалансированные деревья.

В односвязном дереве /3,4/ имеются две специфические особенности: принадлежность объектных узлов только терминальному уровню и равенство наибольшего значения признака среди узлов пучка каждого из уровней значению признака корневого узла пучка. Идентификаторы узлов в пучке упорядочены (слева направо) по возрастанию. При поиске проверяются ключи последовательных узлов пучка и переход к пучку следующего уровня происходит из первого встретившегося узла, идентификатор которого больше поискового предписания. По достижении узла с требуемым значением признака происходит спуск по правым равнозначным узлам пучков нижележащих уровней к объектному узлу. Размеры пучков n на всех уровнях предполагаются одинаковыми, поэтому структура дерева полностью определяется парой (m, n), где m - число уровней.

Ниже предпринята модификация описанной структуры за счет отказа от ее специфики с целью предельного насыщения дерева объектным содержанием. Его поисковые свойства исследуются в двух наиболее общих предположениях: каждый узел несет объектную информацию (является объектным) и размеры пучков могут изменяться от уровня к уровню. При этом наибольший из ключей терминальных узлов каждого пучка всегда меньше ключа корневого узла пучка. С точностью до некоторых технических деталей такая структура обеспечивается алгоритмическими средствами односвязного дерева /4/. Помимо лучшего использования памяти благодаря отказу от узлов без объектной информации, такое дерево должно иметь и лучшие поисковые свойства: поиск заканчивается при обнаружении совпадения на любом уровне и не требуется специального спуска к периферии дерева за объектной информацией. Кроме того, как оказывается, снятие ограничения, связанного с фиксированным размером пучка n, создает дополнительный существенный резерв ускорения поиска.

3. Построение математической модели

При обусловленных предположениях структура исследуемого дерева однозначно определяется вектором

$$\vec{Q}(m, n_0, n_1, \ldots, n_{m-1}),$$

где m - количество уровней дерева, n_i - размер пучка узлов i-ого уровня, $i=\overline{0,m-1}$. Эти параметры следует выбрать из условия минимальной трудоемкости поиска, характеристикой которой может служить длина поискового трека l, определяемая числом проверок, необходимых для достижения искомого объектного узла (цели поиска). Но эта величина сильно зависит от положения цели поиска в ряду последовательных объектных узлов дерева, изменяясь в широких пределах по закону "неправильной пилы".

Лемма. Множество всевозможных значений длин поисковых треков объектных узлов наращиваемого признакового дерева представляет собой непрерывную последовательность целых положительных чисел из сегмента $1 \le l \le 1+\sum_{i=0}^{m-1} n_i$.

Доказательство. Рассмотрим последовательность поисковых треков, каждый из которых на единицу длиннее предыдущего и получается из него в результате проверки следующего узла в пучке, полная проверка которого не завершена при движении по предыдущему треку. Если процесс генерирования новых треков обеспечить таким образом, чтобы "заполнение" пучков осуществлялось последовательно от корня дерева к его нижнему уровню, то длины совокупности полученных таким образом треков будут последовательными целыми числами от I до $1+\sum_{i=0}^{m-1} n_i$, и эта последовательность не имеет разрывов.

Предполагая обращения ко всем узлам дерева равновероятными, в качестве усредненной характеристики трудоемкости поиска примем приведенную длину поискового трека как сумму длин треков всех объектных узлов дерева, отнесенную к их общему числу:

$$\Psi(\vec{Q}) = \Phi(\vec{Q}) / N(\vec{Q}). \qquad (3.1)$$

Умножение узлов в наращиваемом дереве происходит лавинообразно: каждый узел i-ого уровня порождает на следующем уровне допол-

нительно n_i узлов, поэтому емкость дерева, определяемая количеством его объектных узлов, будет

$$N(\vec{Q}) = 1 + \sum_{j=0}^{m-1} \prod_{i=0}^{j} n_i \qquad (3.2)$$

Утверждение I. Сумма длин поисковых треков всех объектных узлов наращиваемого сбалансированного признакового дерева с переменным размером пучка равна:

$$\Phi(\vec{Q}) = 1 + \sum_{\kappa=0}^{m-1} \left\{ \prod_{j=0}^{\kappa} n_j \left(1 + \sum_{i=0}^{\kappa} \frac{n_i + 1}{2} \right) \right\} . \qquad (3.3)$$

Доказательство. Все узлы каждого из уровней рассматриваемого дерева являются объектными, поэтому доказательство сводится к вычислению соответствующих сумм длин треков по уровням и их последующему сложению.

Сумму длин треков узлов κ-го уровня определим интегрированием количеств проверок узлов предыдущих уровней, выполняемых для достижения объектных узлов κ-го уровня в соответствии с описанным алгоритмом поиска. В каждой из зон, ограниченных двумя соседними вышележащими уровнями, число проверок, необходимых для покрытия всех узлов κ-го уровня, можно получить, как в [5], в виде произведения трех составляющих: количества пучков в зоне, суммы длин участков треков внутри зонного пучка и числа объектных узлов κ-го уровня, достижимых через каждый участок трека в пучке зоны. Например, для p-ой зоны, заключенной между p-ым и $(p+1)$-м уровнями, будем иметь соответственно:

$$\Phi_p^{\kappa}(\vec{Q}) = \prod_{i=0}^{p-1} n_i \times \frac{n_p + 1}{2} n_p \times \prod_{j=p+1}^{\kappa-1} n_j = \frac{n_p + 1}{2} \prod_{i=0}^{\kappa-1} n_i , \quad 0 \le p \le \kappa - 1 .$$

Здесь второй сомножитель представляет собой сумму арифметической прогрессии с единичной разностью, которую образуют все возможные количества проверок узлов в зонном пучке. Заметим еще, что трек каждого объектного узла κ-го уровня начинается с проверки идентификатора корня дерева, что объясняет наличие второго слагаемого в выражении для $\Phi^{\kappa}(\vec{Q})$:

$$\Phi^{\kappa}(\vec{Q}) = \sum_{p=0}^{\kappa-1} \Phi_p^{\kappa}(\vec{Q}) + \prod_{i=0}^{\kappa-1} n_i .$$

Теперь искомую сумму определим заключительным интегрированием по уровням. С учетом корня получим:

$$\Phi(\vec{Q}) = 1 + \sum_{\kappa=1}^{m} \Phi^{\kappa}(\vec{Q}) = 1 + \sum_{\kappa=0}^{m-1} \left\{ \prod_{j=0}^{\kappa} n_j \left(1 + \sum_{i=0}^{\kappa} \frac{n_i + 1}{2} \right) \right\}$$

Утверждение доказано.

Теперь задача оптимизации структуры наращиваемого дерева в смысле (3.I) сводится к нахождению вектора $\vec{Q}^*$ такого, чтобы

$$\Psi(\vec{Q}^*) = \min_{\vec{Q} \in \Omega} \frac{\sum_{\kappa=0}^{m-1} \left\{ \prod_{j=0}^{\kappa} n_j \left(1 + \sum_{i=0}^{\kappa} \frac{n_i + 1}{2} \right) \right\} + 1}{\sum_{j=0}^{m-1} \prod_{i=0}^{j} n_i + 1} , \qquad (3.4)$$

где $$\Omega = \left\{ \vec{Q} \,\middle|\, E \le N(\vec{Q}) \le \delta E \right\}, \qquad (3.5)$$

E - прогнозируемое потребное число объектных узлов дерева,

$\delta \ge 1$ - допуск на превышение E, разрешенный наличным резервом памяти,

а компоненты вектора $\vec{Q}$ должны удовлетворять естественным требованиям целочисленности, причем

$$n_i \ge 2 , \quad i = \overline{0, m-1} , \qquad (3.6)$$

$$1 \le m \le \log_2(\delta E) . \qquad (3.7)$$

Это типичная задача целочисленного программирования с ограничениями и сильно выраженной нелинейностью. Общих аналитических приемов для точного решения подобных задач пока не существует, хотя иногда удается найти упрощающие подходы, которые позволяют получить приближенные решения квазиклассическими средствами. Способы установления таких возможностей, исключающих или сводящих к минимуму перебор, не являются общими.

Например, в [5] подобная задача возникла при рассмотрении примера оптимизации на дереве, представляющем собой развитие [4]. В результате некоторого перераспределения терминальных объектных узлов (и только их) по всей структуре дерева они были приближены к корню, что увеличило быстродействие алгоритмов доступа к данным. На основе учета структурных особенностей возникшей дискретной задачи установлено, что условием оптимальности является редукция

размеров пучков от корня к периферии дерева на представительном множестве трех номиналов $S\{4,3,2\}$. Перебором подтверждены и представительность этого множества, и факт редукции. Лучшие поисковые свойства такого дерева, "выпуклого" к корню, объясняются тем, что короткие треки представлены в нем в большем количестве, чем в равноемких деревьях другой структуры.

В смысле сделанного отступления для задачи (3.4)-(3.7) не удалось получить подобных оценок, позволяющих заменить область (3.5) множеством допустимых векторов $\vec{q}$ значительно меньшей мощности. Поэтому для точного решения был разработан алгоритм, реализующий комбинаторную схему направленного перебора по методу ветвей и границ /6/. Результаты машинных экспериментов представлены в следующем параграфе, здесь же мы построим еще композиционную модель рассматриваемого дерева - с целью получить удобный аппарат для наглядной иллюстрации метрических свойств исследуемых деревьев любой структуры.

Утверждение 2. Функция распределения числа поисковых треков по их длинам в наращиваемом сбалансированном признаковом дереве с переменным размером пучка определяется энумератором:

$$F(t) = t\left(1 + \sum_{\kappa=1}^{m-1} \prod_{j=0}^{\kappa} \sum_{i=1}^{n_j} t^i\right) = \sum_{l=1}^{\lambda+1} C_l t^l, \qquad (3.8)$$

где $\lambda = \sum_{i=0}^{m-1} n_i$,

C_l - количество треков длины l;
t - свободный параметр.

Доказательство. В комбинаторном анализе композицией называется кортеж целых положительных чисел при заданной их сумме, который определяется: верхними границами этих чисел, фактическими значениями чисел и характеристикой (их суммой). Можно показать /7/, что производящая перечисляющая функция для композиции с q частями, каждая из которых не превышает h_i, $i=\overline{1,q}$, имеет вид:

$$\prod_{j=1}^{q} \sum_{i=1}^{h_i} t^i = \sum_{z=q}^{\varkappa} M_z t^z,$$

где M_z - количество всевозможных композиций с характеристикой z, $\varkappa = \sum_{j=1}^{q} h_j$.

При поиске объектных узлов κ-ого уровня наращиваемого дерева на каждом из κ уровней количество выполняемых сравнений $d_i \in [1, n_i]$, $i=\overline{0,\kappa-1}$. Кроме того, одна проверка нужна при каждом поиске для сравнения с идентификатором корня дерева. Поэтому в рамках соответствия: $h_i \to n_i$, $(q+1) \to \kappa$, $z \to l$ - имеет место полная аналогия между моделью поиска узлов κ-ого уровня и композицией с $h_i = 1$. Функция распределения числа треков узлов κ-го уровня по их длинам запишется в виде:

$$F_\kappa(t) = t \prod_{j=0}^{\kappa-1} \sum_{i=1}^{n_j} t^i = \sum_{l=\kappa+1}^{\varkappa_\kappa+1} C_l t^l,$$

где $\varkappa_\kappa = \sum_{i=0}^{\kappa-1} n_i$

Заметим еще, что энумератор для нулевого уровня есть

$$F_0(t) = t,$$

т.е. имеется единственный трек - единичной длины. Теперь интегрированием по уровням для всего дерева получим:

$$F(t) = \sum_{\kappa=0}^{m} F_\kappa(t) = t\left(1 + \sum_{\kappa=0}^{m-1} \prod_{j=0}^{\kappa} \sum_{i=1}^{n_j} t^i\right) = \sum_{l=1}^{\lambda+1} C_l t^l.$$

Утверждение доказано.

4. Результаты экспериментов

Для комбинаторного решения задачи (3.4)-(3.7) была составлена программа, реализующая схему одностороннего ветвления по методу ветвей и границ /6/ и существенно использующая для сокращения перебора структурные особенности задачи. Заметим, что превышение свободного резерва памяти над задаваемым E, практически всегда обеспеченное реальными условиями синтеза, частично используется для размещения дерева, т.к. в силу целочисленности размеров пучков в отношении $\nu(\vec{q}) \geq E$ наиболее вероятно неравенство.

В таблице I представлены некоторые результаты машинного эксперимента по синтезу наращиваемых поисковых деревьев для различных значений E.

ТАБЛИЦА I

$E=2^{\kappa}$	$N(\vec{Q}^*)$	$\vec{Q}^*$		$\varphi(\vec{Q}^*)$
κ		m	$n_0, n_1, \ldots, n_{m-1}$	
8	I69	5	4,3,3,2,2	9,335
9	557	6	4,3,3,2,2,2	I0,7I3
I0	I077	6	4,4,3,3,2,2	II,825
II	2229	7	4,4,3,3,2,2,2	I3,207
I2	4309	7	4,4,4,3,3,2,2	I4,32I
I3	89I7	8	4,4,4,3,3,2,2,2	I5,706
I4	I6402	9	3,3,3,3,3,3,3,2,2	I6,80I
I5	33898	I0	3,3,3,3,3,3,3,2,2,2	I8,I94
I6	65609	I0	4,3,3,3,3,3,3,3,2,2	I9,300
I7	I3I593	II	4,3,3,3,3,3,3,3,2,2,2	20,694

Из таблицы I видно, что область значений размеров пучков в оптимальных структурах ограничена множеством $S\{4,3,2\}$ и что, как и в /5/, имеет место эффект редукции размеров пучков от корня к терминальному уровню. Экстраполяция этого результата из /5/ на задачу (3.4)-(3.7) могла прогнозироваться в силу содержательной оправданности геометрической модели дерева, "расширяющегося" к корню: возрастание числа узлов, приближенных к корню, приводит к увеличению количества коротких поисковых треков.

Заметим еще, что изредка (в наших экспериментах 2 случая из 73) среди компонентов вектора $\vec{Q}^*$ появляется $n_0=5$, что не меняет значения сформулированного результата и подтверждает "флуктуационную" природу дискретности. В этих редких случаях значения целевой функции $\varphi(\vec{Q})$ на соседних допустимых векторах, представленных в S, отличались от $\varphi(\vec{Q}^*)$ на величины порядка 10^{-3}.

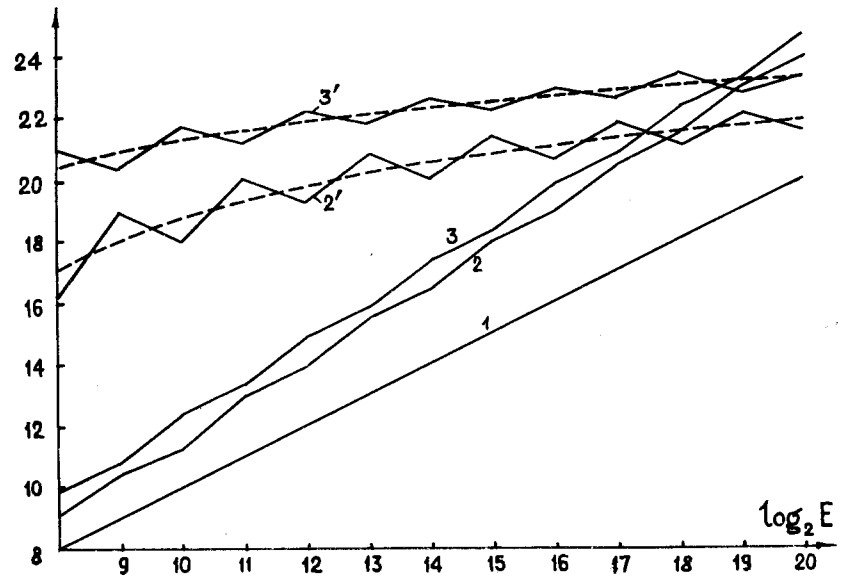

Рис.I.

На рис.I кривые 2 и 3 представляют зависимости $\varphi(\log_2 E)$ соответственно для исследуемого односвязного дерева, предельно насыщенного объектной информацией, и дерева /5/, где число объектных узлов, распределенных по всей структуре, равно количеству терминальных. Легко показать, что в двусвязном сбалансированном дихотомическом дереве $\varphi \approx \log_2 E$ (кривая I). Графики 2′ и 3′ изображают зависимости $(\varphi-\varphi_д)/\varphi_д$ (%) для сравнения с дихотомическим поиском. Они показывают, что в области практически важных значений E поиск замедляется меньше, чем на 20%, тогда как экономия памяти может быть двукратной.

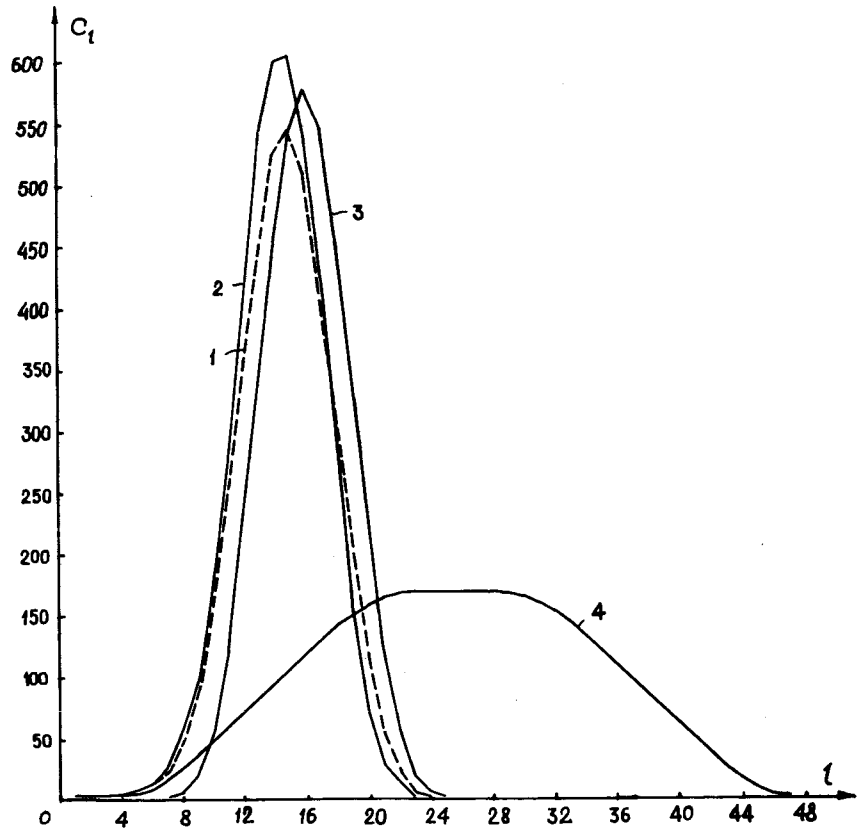

Рис.2.

На рис.2. представлены функции распределения числа поисковых треков по их длинам для деревьев, параметры структуры которых сведены в таблицу 2.

ТАБЛИЦА 2

№	$N(\vec{Q})$	$\vec{Q}$		Коэффициенты знумераторов
		m	$n_0, n_1, \ldots, n_{m-1}$	
I	4I8I	6	4,4,4,4,3,4	I,I,2,4,8,I5,29,55,99,I66,256, 359,457,526,546,5I0,426,3I6, 206,II6,55,2I,6,I.
2	4309	7	4,4,4,3,3,2,2	I,I,2,4,8,I5,29,55,I0I,I75,279, 403,52I,598,607,54I,420,28I, I59,74,27,7,I.
3	4096	6	4,4,4,4,4,4	I,7,22,56,I23,2I7,335,455,548, 580,548,455,335,2I7,I23,56,22, 7,I.
4	42I5	4	2,6,I4,24	I,I,2,3,6,II,28,27,37,48,60,72, 84,96,I08,I20,I32,I43,I52,I59, I64,I67,I68,I68,I68,I68,I68, I68,I67,I64,I59,I52,I43,I32, I20,I08,96,84,72,60,48,36,25, I6,9,4,I.

Дискретный характер задачи затрудняет получение в точности равноемких структур; деревья, представленные в таблице, синтезированы при E=4096. Симметричная кривая[3] соответствует дереву Ландауэра, остальные - структуре (3.8). Из рис.2 видно, что в дереве 2, "выпуклом" к корню, количество коротких треков значительно больше, чем во всех остальных случаях. В структурах, где "выпуклость" выражена слабее, происходит перераспределение числа треков в область больших значений длин, что ухудшает их поисковые свойства (кривая I). Пример 4 иллюстрирует резкое ухудшение поисковых свойств в деревьях с возрастающим от номера уровня размером пучков ("вогнутые" структуры).

ЛИТЕРАТУРА

I. D.E.Knuth. The Art of Computer Programming, Vol.3: Sorting and Searching. Addison-Wesley,Reading,Mass., 1973, 406-479.

2. С.С.Лавров, Л.И.Гончарова. Автоматическая обработка данных. М., "Наука", 1971.

3. А.И.Китов. Программирование экономических и управленческих задач. М., "Советское радио", 1971.

4. W.I.Landauer. The Balanced Tree and its Utilization in Information Retrieval. - IEEE Transactions on EC, 1963, EC-12, №6, December, 863-871.

5. В.М.Сумароков. Некоторые комбинаторные свойства одного класса ассоциативных признаковых структур. Программирование, 1976, № 2, 19-28.

6. A.M.Goeffrion , R.E.Marsten. Integer Programming Algorithms: a Framework and State-of-the-Art Survey. - Management Science, 1972, 18, N 9, 465-491.

7. Дж.Риордан. Введение в комбинаторный анализ. М., "Изд-во иностранной литературы", 1963.

Рукопись поступила в издательский отдел
7 мая 1976 года.