О функциональной сложности двумерной задачи о доминировании

Гасанов Э.Э.

В работе исследуются двумерная задача о доминировании, в которой база данных представляет собой множество точек на плоскости, и надо для произвольной точки плоскости, интерпретируемой как запрос на поиск, найти все точки из базы данных, которые по обоим координатам превосходят запрос. Под функциональной сложностью понимается функция зависимости времени поиска от объема памяти, которую можно выделить под структуры данных. В работе показано, как, используя метод Бентли-Маурера, можно получить алгоритмы с разными соотношениями времени поиска и объема памяти.

Ключевые слова: информационно-графовая модель данных, двумерная задача о доминировании, функциональная сложность, метод сеток.

1. Введение

Для оценки алгоритмов поиска в базах данных обычно используются три характеристики: количество памяти, требуемой под структуры данных, время поиска в худшем случае, и время поиска в среднем. При этом алгоритмы, которые могут обеспечить быстрое время поиска, как правило требуют много памяти для хранения своих структур данных. Поэтому всегда желательно иметь серию различных алгоритмов решения одной и той же задачи поиска с разными соотношениями времени поиска и объема памяти. Тогда разработчик базы данных, зная свои ограничения по памяти, сможет подобрать самый быстрый алгоритм, удовлетворяющий этим требованиям.

Известно, что если задача поиска такая, что на элементах базы данных можно задать линейный порядок, ассоциированный с отношением поиска, то такие задачи можно как правило решать быстро без существенных затрат по памяти, но когда отношение поиска, определяющее, когда элементы базы данных удовлетворяют запросам на поиск, является отношением частичного порядка, тогда для получения хороших характеристик по времени поиска необходимо затратить существенные объемы памяти.

В работе [1] Бентли и Маурером для решения задачи интервального поиска был предложен многоэтапный метод прямого доступа — метод сеток. Он позволяет снижать порядок требуемой памяти, но при этом возрастает время поиска. Этот метод может быть использован и для других задач информационного поиска.

Двумерная задача о доминировании одна из самых простых задач с отношением поиска, являющимся отношением частичного порядка, и на ней идею метода Бетли-Маурера можно продемонстрировать наиболее просто.

В двумерной задаче о доминировании элементы базы данных и запросы на поиск являются точками евклидовой плоскости. Задача состоит в том, чтобы для произвольного запроса на поиск перечислить все элементы базы данных, которые по обоим координатам не меньше, чем запрос.

Скажем, что алгоритм поиска является $(f(k), g(k), h(k))$ алгоритмом, где k — число элементов в базе данных, если при $k \to \infty$ объем памяти, требуемый алгоритму, время поиска в худшем случае и среднее время поиска асимптотически не превосходят соответственно величин $f(k)$, $g(k)$ и $h(k)$.

В работе используется информационно-графовая модель данных, которая позволяет строго формально определить как задачу поиска, так и алгоритмы ее решения и ее сложностные характеристики. Среди последних работ с использованием информационно-графовой модели можно можно отметить [2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

В работе с помощью метода Бентли-Маурера получена серия алгоритмов решения двумерной задачи о доминировании, которые являются $(q \cdot k^{1+1/q}/2, 4q + \log_2 k, 4q - 1)$ алгоритмами, где q — некоторый целочисленный параметр.

Автор выражает благодарность В.Б.Кудрявцеву и А.С.Подколзину за поддержку в работе.

2. Основные понятия и формулировка результата

В соответствии с [12, 13] опишем информационно-графовую модель данных.

Пусть X — множество запросов с заданным на нем вероятностным пространством $\langle X, \sigma, \mathbf{P} \rangle$, где σ — алгебра подмножеств множества X, $\mathbf{P}$ — вероятностная мера на σ; Y — множество записей (объектов поиска); ρ — бинарное отношение на $X \times Y$, называемое отношением поиска. Пятерку $S = \langle X, Y, \rho, \sigma, \mathbf{P} \rangle$ будем называть *типом.* Тройку $I = \langle X, V, \rho \rangle$, где V — некоторое конечное подмножество множества Y, в дальнейшем называемое *библиотекой*, будем называть задачей информационного поиска (ЗИП) типа S, и будем считать, что ЗИП $I = \langle X, V, \rho \rangle$ содержательно состоит в перечислении для произвольно взятого запроса $x \in X$ всех тех и только тех записей $y \in V$ таких, что $x \rho y$.

Пусть f — одноместный предикат, определенный на X , то есть $f : X \to \{0, 1\}$. 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 $\chi_{y,\rho} : X \to \{0, 1\}$ такую, что $N_{\chi_{y,\rho}} = O(y, \rho)$ назовем *характеристической функцией* записи y.

Пусть F — множество символов одноместных предикатов, определенных на множестве X, G — множество символов одноместных переключателей, определенных на множестве X. Под переключателем будем понимать функцию, областью значений которой является начальный отрезок натурального ряда. Пару $\mathcal{F} = \langle F, G \rangle$ назовем базовым множеством.

Понятие *информационного графа* (ИГ) над базовым множеством $\mathcal{F} = \langle F, G \rangle$ определяется следующим образом. Берется конечная многополюсная ориентированная сеть. В ней выбирается некоторый полюс, который называется корнем. Остальные полюса называются листьями и им приписываются записи из Y, причем разным листьям могут быть приписаны одинаковые записи. Некоторые вершины сети (в том числе это могут быть и полюса) называются переключательными и им приписываются переключатели из G. Ребра, исходящие из каждой из переключательных вершин, нумеруются подряд, начиная с 1, и называются переключательными ребрами. Ребра, не являющиеся переключательными, называются предикатными и им приписываются предикаты из множества F. Таким образом нагруженную многополюсную ориентированную сеть называем ИГ над базовым множеством $\mathcal{F} = \langle F, G \rangle$.

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

Пусть нам дана ЗИП $I = \langle X, V, \rho \rangle$. Множество $\mathcal{J}_I(x) = \{y \in V : x\rho y\}$ назовем *ответом* ЗИП I на запрос x. Скажем, что ИГ U *решает* ЗИП $I = \langle X, V, \rho \rangle$, если для любого запроса x из X выполнено $\mathcal{J}_U(x) = \mathcal{J}_I(x)$.

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

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

Сложностью ИГ U на запросе x назовем величину

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

где $\mathcal{R}$ — множество вершинИГ U, $\mathcal{P}$ — множество переключательных вершин ИГ U, ψ_β — количество ребер, исходящих из вершины β. $T(U, x)$ равно числу переключателей и предикатов, вычисленных алгоритмом поиска, соответствующим ИГ U на запросе x, т.е. $T(U, x)$ характеризует время поиска на запросе x.

Сложностью ИГ U в худшем случае назовем величину

$$\widehat{T}(U) = \max_{x \in X} T(U, x)$$

$\widehat{T}(U)$ характеризует время поиска в худшем случае.

Скажем, что базовое множество $\mathcal{F}$ *измеримое*, если каждая функция из $\mathcal{F}$ — измеримая (относительно алгебры σ).

В [13, Лемма 17] доказана справедливость следующей леммы.

Лемма 1. *Если базовое множество $\mathcal{F}$ измеримое, то для любого ИГ U над базовым множеством $\mathcal{F}$ функция $T(U, x)$, как функция от x, является случайной величиной.*

Далее всюду будем предполагать, что базовое множество измеримое, и тем самым $T(U, x)$ будет случайной величиной.

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

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

Через $\mathcal{U}(I,\mathcal{F})$ обозначим множество всех ИГ над базовым множеством $\mathcal{F}$, решающих ЗИП I.

Согласно [13, Теорема 17] для любой ЗИП I, любого ИГ $U \in \mathcal{U}(I,\mathcal{F})$ и любого запроса $x \in X$ выполнена мощностная нижняя оценка

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

Поэтому введем следующие сложностные характеристики ИГ:

- $T'(U,x) = T(U,x) - |\mathcal{J}_I(x)|$, которая характеризует время поиска на запросе x без времени перечисления ответа;
- $\widehat{T}'(U) = \max_{x \in X} T'(U,x)$, которая характеризует время поиска в худшем случае без учета времени перечисления ответа:
- $T'(U) = \mathbf{M}_x T'(U,x)$, которая характеризует среднее время поиска без учета времени перечисления ответа.

На множестве $[0,1]^2 \times [0,1]^2$ введем отношение "$\leq$"следующим образом:

$$(x^1,x^2) \leq (y^1,y^2) \Longleftrightarrow x^1 \leq y^1 \& x^2 \leq y^2.$$

Пусть $Y = (0,1] \times [0,1]$ — множество записей; $V = \{y_1,\ldots,y_k : y_i = (y_i^1,y_i^2),\ i = 1,\ldots,k\} \subseteq Y$ — библиотека; $X = (0,1] \times [0,1]$ — множество запросов; отношение поиска ρ задается на $X \times Y$ следующим соотношением:

$$x \rho y \Longleftrightarrow x \leq y,$$

где $x = (x^1,x^2)$, $y = (y^1,y^2)$, т.е. $x \rho y \Longleftrightarrow x \in (0,y^1] \times [0,y^2]$. Тогда ЗИП $I = \langle X,V,\rho \rangle$ называется двумерной задачей о доминировании.

Здесь у квадратов X и Y выброшена левая сторона только в целях удобства дальнейшего изложения.

Определим переключатели и предикаты

$$f_a(x^1,x^2) = \begin{cases} 1, & \text{если } x^2 \leq a \\ 0, & \text{в противном случае} \end{cases},\ a \in [0,1], \tag{1}$$

$$g_a(x^1,x^2) = \begin{cases} 1, & \text{если } x^1 \leq a \\ 2, & \text{в противном случае} \end{cases},\ a \in [0,1], \tag{2}$$

$$g_{\cdot,m}(x^1, x^2) =](x^1) \cdot m[, \;\; m \in \mathbb{N}, \tag{3}$$

где $\mathbb{N}$ — множество натуральных чисел, $]a[$ — наименьшее целое, не меньшее, чем a.

Пусть

$$F = \{f_a(x^1, x^2) : a \in [0, 1]\}, \tag{4}$$

$$G_1 = \{g_a(x^1, x^2) : a \in [0, 1]\}, \tag{5}$$

$$G_2 = \{g_{\cdot,m}(x^1, x^2) : m \in \mathbb{N}\}, \tag{6}$$

$$\mathcal{F} = \langle F, G_1 \cup G_2 \rangle - \tag{7}$$

базовое множество

Будем писать $A(n) \precsim B(n)$ при $n \to \infty$, если существует такая положительная константа c, что, начиная с некоторого номера n_0, $A(n) \leq c \cdot B(n)$. Если $A \precsim B$ и $B \precsim A$, то будем говорить, что A и B *равны по порядку* при $n \to \infty$ и обозначать $A = O(B)$.

Справедлива следующая теорема.

Теорема 1. *Пусть $I_k = \langle X, V_k, \rho \rangle$ — произвольная последовательность двумерных задач о доминировании, где $|V_k| = k$, $\mathcal{F}$ — базовое множество, определяемое соотношениями (1)–(7). Пусть на X задано вероятностное пространство $\langle X, \sigma, \mathbf{P} \rangle$, и функция $\mathbf{P}$ — равномерная вероятностная мера. Тогда для любого натурального $q \leq \log_2 k$ существует ИГ $U^q_{V_k} \in \mathcal{U}(I_k, \mathcal{F})$, такой что при $k \to \infty$*

$$Q(U^q_{V_k}) = \frac{q}{2} \cdot k^{1+\frac{1}{q}} + O(k),$$

$$\widehat{T}'(U^q_{V_k}) \leq 4q - 1 + \log_2 k,$$

$$T'(U^q_{V_k}) \leq 4q - 1.$$

3. Вспомогательные структуры

Предположим, что библиотека $V = \{y_1, \ldots, y_k : y_i = (y_i^1, y_i^2), \; i = 1, \ldots, k\}$ и запрос $x = (x^1, x^2)$ такие, что мы точно знаем, что $x^1 \leq y_i^1$ для любого $i \in \{1, \ldots, k\}$. Тогда нам для данного запроса x фактически надо решать одномерную задачу о доминировании, т.е. осуществлять проверку только по второй координате.

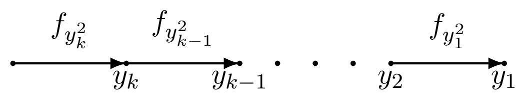

Рис. 1. Граф C_V

Без ограничения общности можно считать, что записи в библиотеке V упорядочены в порядке возрастания вторых координат, т.е. $y_1^2 < y_2^2 < \ldots < y_k^2$.

Согласно [12, Теорема 35] оптимальным алгоритмом для этой задачи будет следующий: просматриваем числа y_i^2, начиная с самого большого и далее в порядке убывания и сравниваем с x^2. Пока $x^2 \leq y_i^2$ запись y_i включаем в ответ, если $x^2 > y_i^2$ или просмотрены все точки библиотеки, то поиск прекращается. Тем самым, время работы этого алгоритма отличается от времени перечисления ответа не более чем на 1.

ИГ, соответствующий этому алгоритму, изображен на рисунке 1. Этот ИГ мы будем обозначать через C_V. Очевидно, что

$$\widehat{T}'(C_V) \leq 1. \tag{8}$$

Далее нам понадобится вспомогательный алгоритм, который будет решать первую задачу о близости, описанную в [12, раздел 2.3]. Эта задача состоит в следующем. Дана библиотека $V = \{y_1, \ldots, y_k : y_i = (y_i^1, y_i^2),\ i = 1, \ldots, k\}$, где записи упорядочены по возрастанию абсцисс, т.е. $y_1^1 < y_2^1 < \ldots < y_k^1$. Для запроса $x = (x^1, x^2)$ В библиотеке V надо найти наименьшую по абсциссе запись, абсцисса которой не меньше, чем x^1.

Дадим неформальное описание алгоритма решения первой задачи о близости.

Разбиваем интервал $(0, 1]$ на k подинтервалов $(\frac{i-1}{k}, \frac{i}{k}]$, $i = 1, 2, \ldots, k$. Разбиваем библиотеку V на части V_i, где в V_i включаются те записи из V, абсциссы которых принадлежат интервалу $(\frac{i-1}{k}, \frac{i}{k}]$.

Пусть на вход алгоритма поступает запрос $x = (x^1, x^2)$. Вычисляем значение переключателя $g_{\cdot,k}(x)$. Пусть $g_{\cdot,k}(x) = i$. Это означает, что $x^1 \in (\frac{i-1}{k}, \frac{i}{k}]$. Если V_i — непустое множество, то бинарным поиском находим в V_i запись, абсцисса которой наименьшая, но не меньшая, чем x^1. Если x^1 больше любой абсциссы записей из V_i или если V_i — пустое множество, то искомая запись есть первая запись в первом непустом множестве V_j таком, что $j > i$.

Согласно [12, Теорема 19] описанное решение имеет константное в среднем время поиска и линейные затраты памяти.

Дадим формальное ИГ D_V, который будет решать первую задачу о близости для множества V и соответствовать описанному выше алгоритму.

Возьмем вершину β_0 и объявим ее корнем графа. Выпустим из β_0 k ребер, припишем им числа от 1 до k, объявим β_0 точкой переключения и припишем ей переключатель $g_{\cdot,k}$.

Пусть $B_i = \{y_j^1, y_j^2) \in V : y_j^1 \in (\frac{i-1}{k}, \frac{i}{k}]$, $l_i = |B_i|$, $i = 1, \dots, k$,

Конец ребра с номером i обозначим β_i.

Для всех i таких, что $B_i \neq \emptyset$, выполним следующую процедуру. Выпустим из вершины β_i бинарное сбалансированное дерево D^{B_i} с $l_i + 1$ концевыми вершинами и высоты $]\log_2(l_i + 1)[$.

Здесь и далее *бинарное дерево* — это ориентированное дерево, в каждую вершину которого, кроме корня, входит одно ребро, и из каждой вершины которого, кроме концевых вершин, исходит два ребра; *корень* — это вершина, в которую не входит ни одно ребро; *концевые вершины* — это вершины из которых не исходит ни одно ребро; *высота вершины* — это длина пути от корня к данной вершине; *сбалансированное дерево* — это дерево, высота концевых вершин которого отличается не более чем на 1; *высота дерева* — это максимальная высота концевых вершин дерева.

Сопоставим концевым вершинам этого дерева D^{B_i}, кроме последней (самой правой), слева направо в порядке возрастания элементы из B_i.

Здесь и далее будем представлять дерево в виде некоторой его укладки на плоскости и понимать направления "вправо "влево" как направления на этой плоскости относительно данной укладки.

Для произвольной внутренней вершины β дерева D^{B_i} обозначим через V_β множество чисел, соответствующих концевым вершинам, достижимым из β. Здесь и далее вершина α *достижима* из вершины β, если существует ориентированный путь ведущий из β в α.

Для произвольной внутренней вершины β дерева D^{B_i} обозначим

$$b_\beta = \max_{b \in V_{\beta'}} b,$$

где β' — конец левого ребра, исходящего из β.

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

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

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

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

Полученный граф и будет ИГ D_V.

Обозначим через α_i концевую вершину, которой приписано число y_i^1, $i = 1, \ldots, k$, а через α_{k+1} самую правую концевую вершину в самом правом дереве D^{B_i}. Легко проверить, что функция проводимости между корнем и вершиной α_i $(i = 1, 2, \ldots, k+1)$ имеет вид:

$$h_i(x^1, x^2) = \begin{cases} 1, & \text{если } y_{i-1}^1 < x^1 \leq y_i^1, \\ 0, & \text{в противном случае,} \end{cases}$$

где $y_0^1 = 0$, $y_{k+1}^1 = 1$.

Подсчитаем объем графа U_m^1.

Поскольку каждое из D^{B_i} есть дерево, в котором полустепень исхода любой вершины равна 2, то в D^{B_i} будет ровно $2 \cdot (l_i + 1) - 2 = 2\, l_i$ ребер. Отсюда следует, что

$$Q(D_V) = k + 2 \sum_{i=1}^{k} l_i = 3\, k. \tag{9}$$

Согласно [12, Теорема 19]

$$\widehat{T}(D_V) \leq 1 +]\log_2(k+1)[, \tag{10}$$

$$T(D_V) \leq 2. \tag{11}$$

4. Доказательство теоремы 1

Доказывать теорему 1 будем индукцией по параметру q.

Базис индукции: $q = 1$.

Пусть $V_k = \{y_1, \ldots, y_k : y_i = (y_i^1, y_i^2),\ i = 1, \ldots, k\}$, причем записи упорядочены по возрастанию абсцисс, т.е. $y_1^1 < y_2^1 < \ldots < y_k^1$.

Обозначим $Z_i^1 = \{y_j : j = i, i+1, \ldots, y_k\}$, $i = 1, 2, \ldots, k$.

Построим ИГ $U_{V_k}^1$.

Возьмем описанный ранее граф D_{V_k}. Корень этого графа объявим корнем графа $U^1_{V_k}$.

Как и ранее, обозначим через α_i концевую вершину ИГ D_{V_k}, которой приписано число y^1_i, $i = 1, \ldots, k$. Для каждого $i = 1, \ldots, k$ из вершины α_i выпустим цепь $C_{Z^1_i}$, которую мы описывали в предыдущем разделе и пример которой приведен на рисунке 1.

Полученный ИГ и будет графом $U^1_{V_k}$.

Убедимся, что этот граф решает задачу $I_k = \langle X, V_k, \rho\rangle$.

В самом деле, если запрос $x = (x^1, x_2)$ попал в вершину α_i, то это означает, что $y^1_{i-1} < x^1 \leq y^1_i$. Что в свою очередь означает, что записи из $V_k \setminus Z^1_i$ не удовлетворяют запросу x, а все записи из Z^1_i подходят по абсциссе. А именно в этих условиях цепь $C_{Z^1_i}$ позволяет решить задачу I_k.

Подсчитаем сложностные характеристики ИГ $U^1_{V_k}$, учитывая соотношения (8) - (11) и, что запрос x может выйти только к одной из вершин α_i:

$$Q(U^1_{V_k}) = Q(D_{V_k}) + \sum_{i=1}^{k} Q(C_{Z^1_i}) = 3\,k + \sum_{i=1}^{k} i = k^2/2 + O(k),$$

$$\widehat{T}'(U^1_{V_k}) \leq \widehat{T}(D_{V_k}) + 1 \leq 1 +]\log_2(k+1)[+ 1 \leq 3 + \log_2 k,$$

$$T'(U^1_{V_k}) \leq T(D_{V_k}) + 1 \leq 3.$$

Тем самым базис индукции доказан.

Индуктивный переход. Предположим мы построили ИГ $U^{q-1}_{V_k}$, решающий ЗИП I_k и имеющий требуемые в условиях теоремы характеристики.

Чтобы в тяжелых технических выкладках не потонула идея метода Бентли-Маурера, мы в дальнейших рассуждениях допустим некоторую вольность и будем рассматривать, вообще говоря, натуральные числа как вещественные.

Пусть

$$p(q,i) = i \cdot k^{1-\frac{1}{q}},\ i = 0, 1, \ldots, k^{\frac{1}{q}},$$

$$R^q = \{y_{p(q,i)},\ i = 1, \ldots, k^{\frac{1}{q}}\},$$

$$W^q_i = \{y_j : p(q, i-1) + 1 \leq j \leq p(q,i),\ i = 1, \ldots, k^{\frac{1}{q}}\},$$

$$Z^q_i = \cup_{j=i+1}^{k^{\frac{1}{q}}} W^q_j,\ i = 1, \ldots, k^{\frac{1}{q}} - 1.$$

Отметим, что $|R^q| = k^{\frac{1}{q}}$, $|W^q_i| = k^{1-\frac{1}{q}}$, $|Z^q_i| = (k^{\frac{1}{q}} - 1 - i) \cdot k^{1-\frac{1}{q}}$.

Построим ИГ $U^q_{V_k}$.

Возьмем описанный ранее граф D_{R^q}. Корень этого графа объявим корнем графа $U^q_{V_k}$.

Обозначим через α_i концевую вершину ИГ D_{R^q}, которой приписано число $y^1_{p(q,i)}$, $i = 1, \ldots, k^{\frac{1}{q}}$. Для каждого $i = 1, \ldots, k^{\frac{1}{q}}$ из вершины α_i выпустим предикатное ребро, которому припишем предикат тождественная 1, а из конца этого ребра выпустим ИГ $U^{q-1}_{W^q_i}$. Кроме того для каждого $i = 1, \ldots, k^{\frac{1}{q}} - 1$ из вершины α_i выпустим цепь $C_{Z^q_i}$, которую мы описывали в предыдущем разделе и пример которой приведен на рисунке 1.

Полученный ИГ и будет графом $U^q_{V_k}$.

Убедимся, что этот граф решает задачу I_k.

В самом деле, если запрос $x = (x^1, x_2)$ попал в вершину α_i, то это означает, что $y^1_{p(q,i-1)} < x^1 \leq y^1_{p(q,i)}$. Что в свою очередь означает, что записи из $\cup_{j=1}^{i-1} W^q_j$ не удовлетворяют запросу x, а все записи из Z^1_i подходят по абсциссе, поэтому из α_i выпускается цепь $C_{Z^q_i}$. А для записей из W^q_i нам надо решать полноценную задачу о доминировании, что и делается с помощью ИГ $U^{q-1}_{W^q_i}$.

Подсчитаем сложностные характеристики ИГ $U^1_{V_k}$, учитывая соотношения (8) - (11) и предположение индукции:

$$Q(U^q_{V_k}) = Q(D_{R^q}) + \sum_{i=1}^{k^{\frac{1}{q}}} Q(U^{q-1}_{W^q_i}) + \sum_{i=1}^{k^{\frac{1}{q}}-1} Q(C_{Z^1_i}) =$$

$$= 3\,k^{\frac{1}{q}} + k^{\frac{1}{q}} \cdot \big(\frac{q-1}{2} k^{\frac{q-1}{q} \cdot \frac{q}{q-1}} + O(k^{\frac{q-1}{q}})\big) + k^{1-\frac{1}{q}} \cdot \sum_{i=1}^{k^{\frac{1}{q}}-1} i \leq$$

$$\leq \frac{q-1}{2} k^{1+\frac{1}{q}} + O(k) + \frac{1}{2} k^{1-\frac{1}{q}} \cdot k^{\frac{2}{q}} = \frac{q}{2} k^{1+\frac{1}{q}} + O(k),$$

$$\widehat{T}'(U^q_{V_k}) \leq \widehat{T}(D_{R^q}) + 1 + \widehat{T}'(U^{q-1}_{V_k}) \leq$$

$$\leq 4 + \frac{1}{q} \log_2 k + 4(q-1) - 1 + \frac{q-1}{q} \log_2 k = 4q - 1 + \log_2 k,$$

$$T'(U^q_{V_k}) \leq T(D_{D_{R^q}}) + 2 + T'(U^{q-1}_{V_k}) \leq 4 + 4(q-1) - 1 = 4q - 1.$$

Тем самым, теорема 1 доказана.

Список литературы

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

[2] Перпер Е.М. Порядок сложности задачи поиска в множестве слов вхождений подслова // Интеллектуальные системы. — 2014. — Т. 19, вып. 1. — С. 99–116.

[3] Плетнев А.А. Информационно-графовая модель динамических баз данных и ее применение// Интеллектуальные системы. — 2014. Т. 18, Вып. 1. — С. 111-140.

[4] Е. М. Перпер. Нижние оценки временной и объемной сложности задачи поиска подслова // Дискретная математика, 2014, том 26:2, 58–70.

[5] Гасанов Э.Э., Ефремов Д.В. Фоновый алгоритм решения двумерной задачи о доминировании // Интеллектуальные системы. — 2014. — Т. 18, вып. 3. — С. 133–158.

[6] Шуткин Ю.С. Моделирование схемных управляющих систем // Интеллектуальные системы. — 2014. — Т. 18, вып. 3. — С. 253–261.

[7] Плетнев А.А. Динамическая база данных, допускающая параллельную обработку произвольных потоков запросов // Интеллектуальные системы. — 2015. Т. 19, Вып. 1. — С. 117–145.

[8] Плетнев А.А. Логарифмическая по сложности параллельная обработка автоматами произвольных потоков запросов в динамической базе данных // Интеллектуальные системы. — 2015. Т. 19, Вып. 1. — С. 171–213.

[9] Плетнев А.А. Нижняя оценка на область видимости автомата, обрабатывающего произвольный поток запросов к динамической базе данных // Интеллектуальные системы. — 2015. Т. 19, Вып. 4. — С. 117–151.

[10] Плетнев А.А. Минимально возможный по степени ветвления информационный граф с радиусом видимости один, обрабатывающий произвольный поток запросов к динамической базе данных // Интеллектуальные системы. — 2016. Т. 20, Вып. 1. — С. 223–254.

[11] Гасанов Э.Э., Плетнев А.А. Моделирование динамических баз данных // Интеллектуальные системы. — 2016. Т. 20, Вып. 3. — С. 146–150.

[12] Гасанов Э. Э, Кудрявцев В. Б. Интеллектуальные системы. Теория хранения и поиска информации. — М.: Юрайт, 2016.

[13] Кудрявцев В. Б, Гасанов Э. Э, Подколзин А. С. Основы теории интеллектуальных систем. — М.: МАКС Пресс, 2016.

On the functional complexity of the two-dimensional domination problem

Gasanov E.E.

In this paper we study the two-dimensional domination problem in which the database is a set of points on the plane, and it is necessary for an arbitrary point of the plane, interpreted as a search query, find all the points from the database, which exceed the query by both coordinates. Functional complexity is the function of the dependence of search time on the memory size. The paper shows how, using the Bentley-Maurer method, we can obtain algorithms with different search time and memory size ratio.

Keywords: information-graph data model, two-dimensional domination problem, functional complexity, grid method.